Spartan-6 Libraries Guide for HDL Designs

UG615 (v 14.1) April 24, 2012

XILINX®

Notice of Disclaimer

XILINX®

Chapter 1

Introduction

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

This guide contains the following:

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

About Design Elements

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

Design elements are divided into three main categories:

- **Macros -** These elements 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.
- **Primitives -** Xilinx components that are native to the FPGA you are targeting. If you instantiate a primitive in your design, after the translation process (ngdbuild) 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.

CORE Generator 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, this guide is one in a series of architecture-specific libraries.

Design Entry Methods

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

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

XILINX®

Chapter 2

About Unimacros

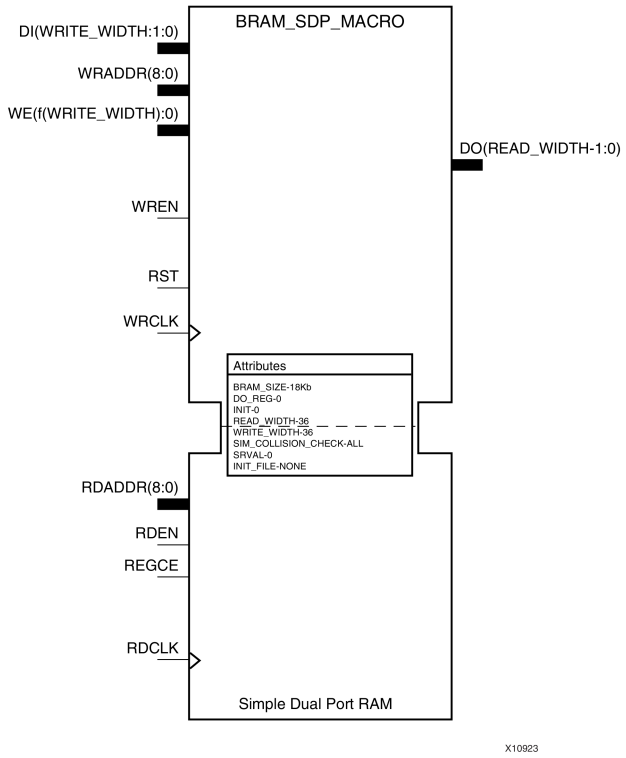
This section describes the unimacros that can be used with this architecture. The unimacros are organized alphabetically.

The following information is provided for each unimacro, where applicable:

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

BRAM_SDP_MACRO

Macro: Simple Dual Port RAM



Introduction

FPGA devices contain several block RAM memories that can be configured as general-purpose 18Kb or 9Kb RAM/ROM memories. These block RAM memories offer fast and flexible storage of large amounts of on-chip data. Both read and write operations are fully synchronous to the supplied clock(s) of the component. However, read and write ports can operate fully independently and asynchronously to each other, accessing the same memory array. Byte-enable write operations are possible, and an optional output register can be used to reduce the clock-to-out times of the RAM.

Note This element, must be configured so that read and write ports have the same width.

Port Description

Name	Direction	Width (Bits)	Function
Output Ports			
DO	Output	See Configuration Table	Data output bus addressed by RDADDR.
Input Ports			
DI	Input	See Configuration Table	Data input bus addressed by WRADDR.
WRADDR, RDADDR	Input	See Configuration Table	Write/Read address input buses.
WE	Input	See Configuration Table	Byte-Wide Write enable.

Name	Direction	Width (Bits)	Function
WREN, RDEN	Input	1	Write/Read enable
SSR	Input	1	Output registers synchronous reset.
REGCE	Input	1	Output register clock enable input (valid only when DO_REG=1)
WRCLK, RDCLK	Input	1	Write/Read clock input.

Configuration Table

DATA_WIDTH	BRAM_SIZE	ADDR	WE
36 - 19	18Kb	9	4
	9Kb	8	
18 - 10	18Kb	10	2
	9Kb	9	
9 - 5	18Kb	11	1
	9Kb	10	
4 - 3	18Kb	12	1
	9Kb	11	
2	18Kb	13	1
	9Kb	12	
1	18Kb	14	1
	9Kb	13	

Design Entry Method

This unimacro can be instantiated only. It is a parameterizable version of the primitive. Consult the Configuration Table above to correctly configure it to meet your design needs.

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

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
BRAM_SIZE	String	"18Kb", "9Kb"	"9Kb"	Configures RAM as "18Kb" or "9Kb" memory.
DO_REG	Integer	0, 1	0	A value of 1 enables to the output registers to the RAM enabling quicker clock-to-out from the RAM at the expense of an added clock cycle of read latency. A value of 0 allows a read in one clock cycle but will have slower clock to out timing.

Attribute	Data Type	Allowed Values	Default	Description
INIT	Hexadecimal	Any 72-Bit Value	All zeros	Specifies the initial value on the output after configuration.
READ_WIDTH, WRITE_WIDTH	Integer	1-72	36	Specifies the size of the DI and DO buses. The following combinations are allowed: • READ_WIDTH = WRITE_WIDTH • If asymmetric, READ_WIDTH and WRITE_WIDTH must be in the ratio of 2, or must be values allowed by the unisim (1, 2, 4, 8, 9, 16, 18, 32, 36, 64, 72)
INIT_FILE	String	String representing file name and location	None	Name of the file containing initial values.
SIM_COLLISION_CHECK	String	"ALL", "WARNING_ONLY", "GENERATE_X_ONLY", "NONE"	"ALL"	Allows modification of the simulation behavior if a memory collision occurs. The output is affected as follows: • "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 go unnoticed during simulation. Care should be taken when changing the value of this attribute. Please see the *Synthesis and Simulation Design Guide* for more information.
SIM_MODE	String	"SAFE", "FAST" .	"SAFE"	This is a simulation only attribute. It will direct the simulation model to run in performance-oriented mode when set to "FAST." Please see the *Synthesis and Simulation Design Guide* for more information.
SRVAL	Hexadecimal	Any 72-Bit Value	All zeroes	Specifies the output value of on the DO port upon the assertion of the synchronous reset (RST) signal.

Attribute	Data Type	Allowed Values	Default	Description
INIT_00 to INIT_7F	Hexadecimal	Any 256-Bit Value	All zeroes	Allows specification of the initial contents of the 16Kb or 32Kb data memory array.
INITP_00 to INITP_0F	Hexadecimal	Any 256-Bit Value	All zeroes	Allows specification of the initial contents of the 2Kb or 4Kb 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;


-- BRAM_SDP_MACRO: Simple Dual Port RAM
--                 Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

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


-----------------------------------------------------------------------
--  READ_WIDTH | BRAM_SIZE | READ Depth  | RDADDR Width |            --
-- WRITE_WIDTH |           | WRITE Depth | WRADDR Width |  WE Width  --
-- ============|===========|=============|==============|============--
--    19-36    |  "18Kb"   |      512    |     9-bit    |    4-bit   --
--    10-18    |  "18Kb"   |     1024    |    10-bit    |    2-bit   --
--    10-18    |   "9Kb"   |      512    |     9-bit    |    2-bit   --
--     5-9     |  "18Kb"   |     2048    |    11-bit    |    1-bit   --
--     5-9     |   "9Kb"   |     1024    |    10-bit    |    1-bit   --
--     3-4     |  "18Kb"   |     4096    |    12-bit    |    1-bit   --
--     3-4     |   "9Kb"   |     2048    |    11-bit    |    1-bit   --
--       2     |  "18Kb"   |     8192    |    13-bit    |    1-bit   --
--       2     |  " 9Kb"   |     4096    |    12-bit    |    1-bit   --
--       1     |  "18Kb"   |    16384    |    14-bit    |    1-bit   --
--       1     |   "9Kb"   |     8192    |    13-bit    |    1-bit   --
-----------------------------------------------------------------------

BRAM_SDP_MACRO_inst : BRAM_SDP_MACRO
generic map (
   BRAM_SIZE => "18Kb", -- Target BRAM, "9Kb" or "18Kb"
   DEVICE => "SPARTAN6", -- Target device: "VIRTEX5", "VIRTEX6", "SPARTAN6"
   WRITE_WIDTH => 0,    -- Valid values are 1-36
   READ_WIDTH => 0,     -- Valid values are 1-36
   DO_REG => 0, -- Optional output register (0 or 1)
   INIT_FILE => "NONE",
   SIM_COLLISION_CHECK => "ALL", -- Collision check enable "ALL", "WARNING_ONLY",
                                 -- "GENERATE_X_ONLY" or "NONE"
   SRVAL => X"0000000000000000", --  Set/Reset value for port output
   INIT => X"0000000000000000", --  Initial values on output port
   -- The following INIT_xx declarations specify the initial contents of the RAM
   INIT_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_07 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_08 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_09 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_0A => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_0B => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_0C => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_0D => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_0E => X"0000000000000000000000000000000000000000000000000000000000000000",
```

```
   INIT_0F => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_10 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_11 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_12 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_13 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_14 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_15 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_16 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_17 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_18 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_19 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_1A => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_1B => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_1C => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_1D => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_1E => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_1F => X"0000000000000000000000000000000000000000000000000000000000000000",

   -- The next set of INIT_xx are for "18Kb" configuration only
   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",

   -- The next set of INITP_xx are for "18Kb" configuration only
   INITP_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_07 => X"0000000000000000000000000000000000000000000000000000000000000000")

port map (
   DO => DO,         -- Output read data port, width defined by READ_WIDTH parameter
   DI => DI,         -- Input write data port, width defined by WRITE_WIDTH parameter
   RDADDR => RDADDR, -- Input read address, width defined by read port depth
   RDCLK => RDCLK,   -- 1-bit input read clock
   RDEN => RDEN,     -- 1-bit input read port enable
   REGCE => REGCE,   -- 1-bit input read output register enable
   RST => RST,       -- 1-bit input reset
   WE => WE,         -- Input write enable, width defined by write port depth
```

```
   WRADDR => WRADDR, -- Input write address, width defined by write port depth
   WRCLK => WRCLK,   -- 1-bit input write clock
   WREN => WREN      -- 1-bit input write port enable
);
-- End of BRAM_SDP_MACRO_inst instantiation
```

Verilog Instantiation Template

```
// BRAM_SDP_MACRO: Simple Dual Port RAM
//                 Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

//////////////////////////////////////////////////////////////////////
//  READ_WIDTH | BRAM_SIZE | READ Depth  | RDADDR Width |            //
// WRITE_WIDTH |           | WRITE Depth | WRADDR Width |  WE Width  //
// ============|===========|=============|==============|============//
//    19-36    |  "18Kb"   |      512    |     9-bit    |    4-bit   //
//    10-18    |  "18Kb"   |     1024    |    10-bit    |    2-bit   //
//    10-18    |   "9Kb"   |      512    |     9-bit    |    2-bit   //
//     5-9     |  "18Kb"   |     2048    |    11-bit    |    1-bit   //
//     5-9     |   "9Kb"   |     1024    |    10-bit    |    1-bit   //
//     3-4     |  "18Kb"   |     4096    |    12-bit    |    1-bit   //
//     3-4     |   "9Kb"   |     2048    |    11-bit    |    1-bit   //
//       2     |  "18Kb"   |     8192    |    13-bit    |    1-bit   //
//       2     |  " 9Kb"   |     4096    |    12-bit    |    1-bit   //
//       1     |  "18Kb"   |    16384    |    14-bit    |    1-bit   //
//       1     |   "9Kb"   |     8192    |    13-bit    |    1-bit   //
//////////////////////////////////////////////////////////////////////

BRAM_SDP_MACRO #(
   .BRAM_SIZE("18Kb"), // Target BRAM, "9Kb" or "18Kb"
   .DEVICE("SPARTAN6"), // Target device: "VIRTEX5", "VIRTEX6", "SPARTAN6"
   .WRITE_WIDTH(0),    // Valid values are 1-36
   .READ_WIDTH(0),     // Valid values are 1-36
   .DO_REG(0),         // Optional output register (0 or 1)
   .INIT_FILE ("NONE"),
   .SIM_COLLISION_CHECK ("ALL"), // Collision check enable "ALL", "WARNING_ONLY",
                                 //   "GENERATE_X_ONLY" or "NONE"
   .SRVAL(72'h000000000000000000), // Set/Reset value for port output
   .INIT(72'h000000000000000000),  // Initial values on output port
   .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),
```

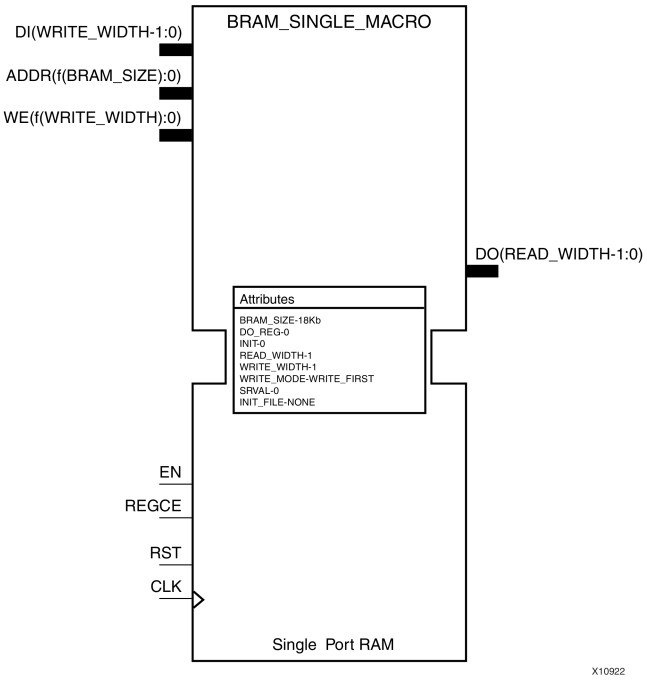
```
   .INIT_1F(256'h0000000000000000000000000000000000000000000000000000000000000000),

   // The next set of INIT_xx are for "18Kb" configuration only
   .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),

   // The next set of INITP_xx are for "18Kb" configuration only
   .INITP_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_05(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_06(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_07(256'h0000000000000000000000000000000000000000000000000000000000000000)
) BRAM_SDP_MACRO_inst (
   .DO(DO),         // Output read data port, width defined by READ_WIDTH parameter
   .DI(DI),         // Input write data port, width defined by WRITE_WIDTH parameter
   .RDADDR(RDADDR), // Input read address, width defined by read port depth
   .RDCLK(RDCLK),   // 1-bit input read clock
   .RDEN(RDEN),     // 1-bit input read port enable
   .REGCE(REGCE),   // 1-bit input read output register enable
   .RST(RST),       // 1-bit input reset
   .WE(WE),         // Input write enable, width defined by write port depth
   .WRADDR(WRADDR), // Input write address, width defined by write port depth
   .WRCLK(WRCLK),   // 1-bit input write clock
   .WREN(WREN)      // 1-bit input write port enable
);

// End of BRAM_SDP_MACRO_inst instantiation
```

For More Information

See the *Spartan-6 FPGA User Documentation (User Guides and Data Sheets)*.

BRAM_SINGLE_MACRO

Macro: Single Port RAM



Introduction

FPGA devices contain several block RAM memories that can be configured as general-purpose 18Kb or 9Kb RAM/ROM memories. These single-port, block RAM memories offer fast and flexible storage of large amounts of on-chip data. Byte-enable write operations are possible, and an optional output register can be used to reduce the clock-to-out times of the RAM.

Port Description

Name	Direction	Width	Function
Output Ports			
DO	Output	See Configuration Table below.	Data output bus addressed by ADDR.
Input Ports			
DI	Input	See Configuration Table below.	Data input bus addressed by ADDR.
ADDR	Input	See Configuration Table below.	Address input bus.
WE	Input	See Configuration Table below.	Byte-Wide Write enable.
EN	Input	1	Write/Read enables.
RST	Input	1	Output registers synchronous reset.
REGCE	Input	1	Output register clock enable input (valid only when DO_REG=1)
CLK	Input	1	Clock input.

Configuration Table

WRITE_WIDTH	BRAM_SIZE	ADDR	WE
37 - 72	18Kb	8	8
36 - 19		9	4
18 - 10		10	2
9 - 5		11	1
4 - 3		12	1
2		13	1
1		14	1
36 - 19	9Kb	8	4
18-10		9	2
9 - 5		10	1
4 - 3		11	1
2		12	1
1		13	1

Design Entry Method

This unimacro can be instantiated only. It is a parameterizable version of the primitive. Consult the above Configuration Table in correctly configuring this element to meet your design needs.

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

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
BRAM_SIZE	String	"18Kb", "9Kb"	"9Kb"	Configures RAM as 18Kb or 9Kb memory.
DO_REG	Integer	0, 1	0	A value of 1 enables to the output registers to the RAM enabling quicker clock-to-out from the RAM at the expense of an added clock cycle of read latency. A value of 0 allows a read in one clock cycle but will have slower clock to out timing.
READ_WIDTH, WRITE_WIDTH	Integer	1 - 36	1	Specifies the size of the DI and DO buses. READ_WIDTH and WRITE_WIDTH must be equal.
INIT_FILE	String	String representing file name and location	NONE	Name of the file containing initial values.
WRITE_MODE	String	"READ_FIRST", "WRITE_FIRST", "NO_CHANGE"	"WRITE_FIRST"	Specifies write mode to the memory

Attribute	Data Type	Allowed Values	Default	Description
INIT	Hexadecimal	Any 72-Bit Value	All zeros	Specifies the initial value on the output after configuration.
SRVAL	Hexadecimal	Any 72-Bit Value	All zeroes	Specifies the output value of on the DO port upon the assertion of the synchronous reset (RST) signal.
SIM_MODE	String	"SAFE", "FAST"	"SAFE"	This is a simulation only attribute. It will direct the simulation model to run in performance-oriented mode when set to "FAST." Please see the *Synthesis and Simulation Design Guide* for more information.
INIT_00 to INIT_FF	Hexadecimal	Any 256-Bit Value	All zeroes	Allows specification of the initial contents of the 16Kb or 32Kb data memory array.
INITP_00 to INITP_0F	Hexadecimal	Any 256-Bit Value	All zeroes	Allows specification of the initial contents of the 2Kb or 4Kb 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;


-- BRAM_SINGLE_MACRO: Single Port RAM
--                    Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

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

-----------------------------------------------------------------------
--  READ_WIDTH | BRAM_SIZE | READ Depth  | ADDR Width |            --
-- WRITE_WIDTH |           | WRITE Depth |            |  WE Width  --
-- ============|===========|=============|============|============--
--    19-36    |  "18Kb"   |      512    |    9-bit   |    4-bit   --
--    10-18    |  "18Kb"   |     1024    |   10-bit   |    2-bit   --
--    10-18    |   "9Kb"   |      512    |    9-bit   |    2-bit   --
--     5-9     |  "18Kb"   |     2048    |   11-bit   |    1-bit   --
--     5-9     |   "9Kb"   |     1024    |   10-bit   |    1-bit   --
--     3-4     |  "18Kb"   |     4096    |   12-bit   |    1-bit   --
--     3-4     |   "9Kb"   |     2048    |   11-bit   |    1-bit   --
--       2     |  "18Kb"   |     8192    |   13-bit   |    1-bit   --
--       2     |   "9Kb"   |     4096    |   12-bit   |    1-bit   --
--       1     |  "18Kb"   |    16384    |   14-bit   |    1-bit   --
--       1     |   "9Kb"   |     8192    |   13-bit   |    1-bit   --
-----------------------------------------------------------------------


BRAM_SINGLE_MACRO_inst : BRAM_SINGLE_MACRO
generic map (
   BRAM_SIZE => "18Kb", -- Target BRAM, "9Kb" or "18Kb"
   DEVICE => "SPARTAN6", -- Target Device: "VIRTEX5", "VIRTEX6", "SPARTAN6"
   DO_REG => 0, -- Optional output register (0 or 1)
   INIT => X"000000000",    --  Initial values on output port
   INIT_FILE => "NONE",
   WRITE_WIDTH => 0,   -- Valid values are 1-36 (19-36 only valid when BRAM_SIZE="18Kb")
   READ_WIDTH => 0,   -- Valid values are 1-36 (19-36 only valid when BRAM_SIZE="18Kb")
   SRVAL => X"000000000",   -- Set/Reset value for port output
   WRITE_MODE => "WRITE_FIRST", -- "WRITE_FIRST", "READ_FIRST" or "NO_CHANGE"
   -- The following INIT_xx declarations specify the initial contents of the RAM
   INIT_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
```

```
INIT_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_07 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_08 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_09 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0F => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_10 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_11 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_12 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_13 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_14 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_15 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_16 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_17 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_18 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_19 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1F => X"0000000000000000000000000000000000000000000000000000000000000000",

-- The next set of INIT_xx are for "18Kb" configuration only
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",

-- The next set of INITP_xx are for "18Kb" configuration only
INITP_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
```

```
   INITP_07 => X"0000000000000000000000000000000000000000000000000000000000000000")

 port map (
   DO => DO,      -- Output data, width defined by READ_WIDTH parameter
   ADDR => ADDR,  -- Input address, width defined by read/write port depth
   CLK => CLK,    -- 1-bit input clock
   DI => DI,      -- Input data port, width defined by WRITE_WIDTH parameter
   EN => EN,      -- 1-bit input RAM enable
   REGCE => REGCE, -- 1-bit input output register enable
   RST => RST,    -- 1-bit input reset
   WE => WE       -- Input write enable, width defined by write port depth
);

-- End of BRAM_SINGLE_MACRO_inst instantiation
```

Verilog Instantiation Template

```
// BRAM_SINGLE_MACRO: Single Port RAM
//                    Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

//////////////////////////////////////////////////////////////////////
//  READ_WIDTH | BRAM_SIZE | READ Depth  | ADDR Width |            //
// WRITE_WIDTH |           | WRITE Depth |            |  WE Width  //
// ============|===========|=============|============|============//
//    19-36    |  "18Kb"   |      512    |    9-bit   |    4-bit   //
//    10-18    |  "18Kb"   |     1024    |   10-bit   |    2-bit   //
//    10-18    |   "9Kb"   |      512    |    9-bit   |    2-bit   //
//     5-9     |  "18Kb"   |     2048    |   11-bit   |    1-bit   //
//     5-9     |   "9Kb"   |     1024    |   10-bit   |    1-bit   //
//     3-4     |  "18Kb"   |     4096    |   12-bit   |    1-bit   //
//     3-4     |   "9Kb"   |     2048    |   11-bit   |    1-bit   //
//       2     |  "18Kb"   |     8192    |   13-bit   |    1-bit   //
//       2     |   "9Kb"   |     4096    |   12-bit   |    1-bit   //
//       1     |  "18Kb"   |    16384    |   14-bit   |    1-bit   //
//       1     |   "9Kb"   |     8192    |   13-bit   |    1-bit   //
//////////////////////////////////////////////////////////////////////

BRAM_SINGLE_MACRO #(
   .BRAM_SIZE("18Kb"), // Target BRAM, "9Kb" or "18Kb"
   .DEVICE("SPARTAN6"), // Target Device: "VIRTEX5", "VIRTEX6", "SPARTAN6"
   .DO_REG(0), // Optional output register (0 or 1)
   .INIT(36'h000000000), // Initial values on output port
   .INIT_FILE ("NONE"),
   .WRITE_WIDTH(0), // Valid values are 1-36 (19-36 only valid when BRAM_SIZE="18Kb")
   .READ_WIDTH(0),  // Valid values are 1-36 (19-36 only valid when BRAM_SIZE="18Kb")
   .SRVAL(36'h000000000), // Set/Reset value for port output
   .WRITE_MODE("WRITE_FIRST"), // "WRITE_FIRST", "READ_FIRST", or "NO_CHANGE"
   .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),
```

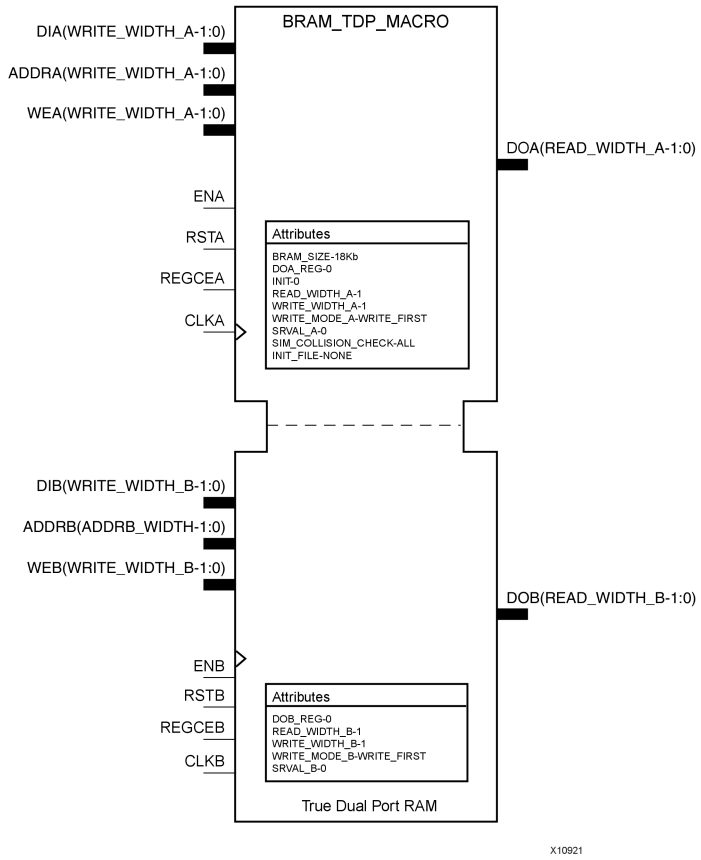
```
   .INIT_17(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_18(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_19(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_1A(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_1B(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_1C(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_1D(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_1E(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_1F(256'h0000000000000000000000000000000000000000000000000000000000000000),

   // The next set of INIT_xx are for "18Kb" configuration only
   .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),

   // The next set of INITP_xx are for "18Kb" configuration only
   .INITP_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_05(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_06(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_07(256'h0000000000000000000000000000000000000000000000000000000000000000)
) BRAM_SINGLE_MACRO_inst (
   .DO(DO),       // Output data, width defined by READ_WIDTH parameter
   .ADDR(ADDR),   // Input address, width defined by read/write port depth
   .CLK(CLK),     // 1-bit input clock
   .DI(DI),       // Input data port, width defined by WRITE_WIDTH parameter
   .EN(EN),       // 1-bit input RAM enable
   .REGCE(REGCE), // 1-bit input output register enable
   .RST(RST),     // 1-bit input reset
   .WE(WE)        // Input write enable, width defined by write port depth
);

// End of BRAM_SINGLE_MACRO_inst instantiation
```

For More Information

See the *Spartan-6 FPGA User Documentation (User Guides and Data Sheets)*.

BRAM_TDP_MACRO

Macro: True Dual Port RAM



Introduction

FPGA devices contain several block RAM memories that can be configured as general-purpose 18kb or 9kb RAM/ROM memories. These block RAM memories offer fast and flexible storage of large amounts of on-chip data. Both read and write operations are fully synchronous to the supplied clock(s) of the component. However, READ and WRITE ports can operate fully independently and asynchronous to each other, accessing the same memory array. Byte-enable write operations are possible, and an optional output register can be used to reduce the clock-to-out times of the RAM.

Port Description

Name	Direction	Width	Function
Output Ports			
DOA	Output	See Configuration Table below.	Data output bus addressed by ADDRA.
DOB	Output	See Configuration Table below.	Data output bus addressed by ADDRB.
Input Ports			
DIA	Input	See Configuration Table below.	Data input bus addressed by ADDRA.

Name	Direction	Width	Function
DIB	Input	See Configuration Table below.	Data input bus addressed by ADDRB.
ADDRA, ADDRB	Input	See Configuration Table below.	Address input buses for Port A, B.
WEA, WEB	Input	See Configuration Table below.	Write enable for Port A, B.
ENA, ENB	Input	1	Write/Read enables for Port A, B.
RSTA, RSTB	Input	1	Output registers synchronous reset for Port A, B.
REGCEA, REGCEB	Input	1	Output register clock enable input for Port A, B (valid only when DO_REG=1)
CLKA, CLKB	Input	1	Write/Read clock input for Port A, B.

Configuration Table

WRITE_WIDTH_A/B-DIA/DIB	BRAM_SIZE	ADDRA/B	WEA/B
36 - 19	18Kb	9	4
18 - 10		10	2
9 - 5		11	1
4 - 3		12	1
2		13	1
1		14	1
18 - 10	9Kb	9	2
9 - 5		10	1
4 - 3		11	1
2		12	1
1		13	1

Design Entry Method

This unimacro can be instantiated only. It is a parameterizable version of the primitive. Consult the Configuration Table above to correctly configure it to meet your design needs.

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

Available Attributes

Attribute(s)	Data Type	Allowed Values	Default	Description
BRAM_SIZE	String	"18Kb", "9Kb"	"9Kb"	Configures RAM as 18Kb or 9Kb memory.
DO_REG	Integer	0, 1	0	A value of 1 enables to the output registers to the RAM enabling quicker clock-to-out from the RAM at the expense of an added clock cycle of read latency. A value of 0 allows a read in one clock cycle but will have slower clock to out timing.
INIT	Hexa-decimal	Any 72-Bit Value	All zeros	Specifies the initial value on the output after configuration.
INIT_FILE	String	String representing file name and location	NONE	Name of file containing initial values.
READ_WIDTH, WRITE_WIDTH	Integer	1 - 72	36	Specifies the size of the DI and DO buses. READ_WIDTH and WRITE_WIDTH must be equal.
SIM_COLLISION_ CHECK	String	"ALL", "WARNING_ ONLY", "GENERATE_X_ ONLY", "NONE"	"ALL"	Allows modification of the simulation behavior if a memory collision occurs. The output is affected as follows: • "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 go unnoticed during simulation. Care should be taken when changing the value of this attribute. Please see the *Synthesis and Simulation Design Guide* for more information.
SIM_MODE	String	"SAFE", "FAST" .	"SAFE"	This is a simulation only attribute. It will direct the simulation model to run in performance-oriented mode when set to "FAST." Please see the *Synthesis and Simulation Design Guide* for more information.
SRVAL_A, SRVAL_B	Hexa-decimal	Any 72-Bit Value	All zeroes	Specifies the output value of on the DO port upon the assertion of the synchronous reset (RST) signal.
INIT_00 to INIT_FF	Hexa-decimal	Any 256-Bit Value	All zeroes	Allows specification of the initial contents of the 16Kb or 32Kb data memory array.
INITP_00 to INITP_0F	Hexa-decimal	Any 256-Bit Value	All zeroes	Allows specification of the initial contents of the 2Kb or 4Kb 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;


-- BRAM_TDP_MACRO: True Dual Port RAM
--                 Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

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

-----------------------------------------------------------------------
-- DATA_WIDTH_A/B | BRAM_SIZE | RAM Depth | ADDRA/B Width | WEA/B Width --
-- ===============|===========|===========|===============|=============--
--     19-36      |  "18Kb"   |     512   |     9-bit     |    4-bit    --
--     10-18      |  "18Kb"   |    1024   |    10-bit     |    2-bit    --
--     10-18      |   "9Kb"   |     512   |     9-bit     |    2-bit    --
--      5-9       |  "18Kb"   |    2048   |    11-bit     |    1-bit    --
--      5-9       |   "9Kb"   |    1024   |    10-bit     |    1-bit    --
--      3-4       |  "18Kb"   |    4096   |    12-bit     |    1-bit    --
--      3-4       |   "9Kb"   |    2048   |    11-bit     |    1-bit    --
--        2       |  "18Kb"   |    8192   |    13-bit     |    1-bit    --
--        2       |   "9Kb"   |    4096   |    12-bit     |    1-bit    --
--        1       |  "18Kb"   |   16384   |    14-bit     |    1-bit    --
--        1       |   "9Kb"   |    8192   |    12-bit     |    1-bit    --
-----------------------------------------------------------------------


BRAM_TDP_MACRO_inst : BRAM_TDP_MACRO
generic map (
   BRAM_SIZE => "18Kb", -- Target BRAM, "9Kb" or "18Kb"
   DEVICE => "SPARTAN6", -- Target Device: "VIRTEX5", "VIRTEX6", "SPARTAN6"
   DOA_REG => 0, -- Optional port A output register (0 or 1)
   DOB_REG => 0, -- Optional port B output register (0 or 1)
   INIT_A => X"000000000", -- Initial values on A output port
   INIT_B => X"000000000", -- Initial values on B output port
   INIT_FILE => "NONE",
   READ_WIDTH_A => 0,   -- Valid values are 1-36
   READ_WIDTH_B => 0,   -- Valid values are 1-36
   SIM_COLLISION_CHECK => "ALL", -- Collision check enable "ALL", "WARNING_ONLY",
                                 -- "GENERATE_X_ONLY" or "NONE"
   SRVAL_A => X"000000000",   -- Set/Reset value for A port output
   SRVAL_B => X"000000000",   -- Set/Reset value for B port output
   WRITE_MODE_A => "WRITE_FIRST", -- "WRITE_FIRST", "READ_FIRST" or "NO_CHANGE"
   WRITE_MODE_B => "WRITE_FIRST", -- "WRITE_FIRST", "READ_FIRST" or "NO_CHANGE"
   WRITE_WIDTH_A => 0, -- Valid values are 1-36
   WRITE_WIDTH_B => 0, -- Valid values are 1-36
   -- 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",

   -- The next set of INIT_xx are for "18Kb" configuration only
   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",

   -- The next set of INITP_xx are for "18Kb" configuration only
   INITP_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_07 => X"0000000000000000000000000000000000000000000000000000000000000000")
 port map (
   DOA => DOA,       -- Output port-A data
   DOB => DOB,       -- Output port-B data
   ADDRA => ADDRA,   -- Input port-A address
   ADDRB => ADDRB,   -- Input port-B address
   CLKA => CLKA,     -- Input port-A clock
   CLKB => CLKB,     -- Input port-B clock
   DIA => DIA,       -- Input port-A data
   DIB => DIB,       -- Input port-B data
   ENA => ENA,       -- Input port-A enable
   ENB => ENB,       -- Input port-B enable
   REGCEA => REGCEA, -- Input port-A output register enable
   REGCEB => REGCEB, -- Input port-B output register enable
   RSTA => RSTA,     -- Input port-A reset
   RSTB => RSTB,     -- Input port-B reset
   WEA => WEA,       -- Input port-A write enable
   WEB => WEB        -- Input port-B write enable
);

-- End of BRAM_TDP_MACRO_inst instantiation
```

Verilog Instantiation Template

```
// BRAM_TDP_MACRO: True Dual Port RAM
//                 Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

//////////////////////////////////////////////////////////////////////
// DATA_WIDTH_A/B | BRAM_SIZE | RAM Depth | ADDRA/B Width | WEA/B Width //
// ===============|===========|===========|===============|=============//
//     19-36      |  "18Kb"   |     512   |     9-bit     |    4-bit    //
//     10-18      |  "18Kb"   |    1024   |    10-bit     |    2-bit    //
//     10-18      |   "9Kb"   |     512   |     9-bit     |    2-bit    //
//      5-9       |  "18Kb"   |    2048   |    11-bit     |    1-bit    //
//      5-9       |   "9Kb"   |    1024   |    10-bit     |    1-bit    //
//      3-4       |  "18Kb"   |    4096   |    12-bit     |    1-bit    //
//      3-4       |   "9Kb"   |    2048   |    11-bit     |    1-bit    //
//        2       |  "18Kb"   |    8192   |    13-bit     |    1-bit    //
//        2       |   "9Kb"   |    4096   |    12-bit     |    1-bit    //
//        1       |  "18Kb"   |   16384   |    14-bit     |    1-bit    //
//        1       |   "9Kb"   |    8192   |    12-bit     |    1-bit    //
//////////////////////////////////////////////////////////////////////

BRAM_TDP_MACRO #(
   .BRAM_SIZE("18Kb"), // Target BRAM: "9Kb" or "18Kb"
   .DEVICE("SPARTAN6"), // Target device: "VIRTEX5", "VIRTEX6", "SPARTAN6"
   .DOA_REG(0),        // Optional port A output register (0 or 1)
   .DOB_REG(0),        // Optional port B output register (0 or 1)
   .INIT_A(36'h0000000),  // Initial values on port A output port
   .INIT_B(36'h00000000), // Initial values on port B output port
   .INIT_FILE ("NONE"),
   .READ_WIDTH_A (0),   // Valid values are 1-36
   .READ_WIDTH_B (0),   // Valid values are 1-36
   .SIM_COLLISION_CHECK ("ALL"), // Collision check enable "ALL", "WARNING_ONLY",
                                 //   "GENERATE_X_ONLY" or "NONE"
   .SRVAL_A(36'h00000000), // Set/Reset value for port A output
   .SRVAL_B(36'h00000000), // Set/Reset value for port B output
   .WRITE_MODE_A("WRITE_FIRST"), // "WRITE_FIRST", "READ_FIRST", or "NO_CHANGE"
   .WRITE_MODE_B("WRITE_FIRST"), // "WRITE_FIRST", "READ_FIRST", or "NO_CHANGE"
   .WRITE_WIDTH_A(0), // Valid values are 1-36
   .WRITE_WIDTH_B(0), // Valid values are 1-36
   .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),
```

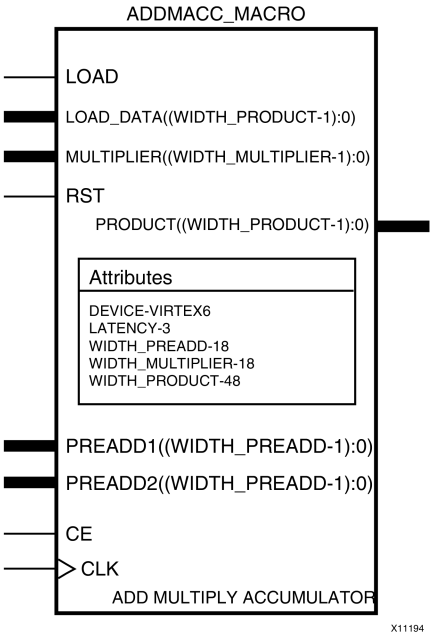
```
   .INIT_1E(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_1F(256'h0000000000000000000000000000000000000000000000000000000000000000),

   // The next set of INIT_xx are for "18Kb" configuration only
   .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
   .INIT_FF(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_03(256'h0000000000000000000000000000000000000000000000000000000000000000),

   // The next set of INITP_xx are for "18Kb" configuration only
   .INITP_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_05(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_06(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_07(256'h0000000000000000000000000000000000000000000000000000000000000000)
) BRAM_TDP_MACRO_inst (
   .DOA(DOA),       // Output port-A data, width defined by READ_WIDTH_A parameter
   .DOB(DOB),       // Output port-B data, width defined by READ_WIDTH_B parameter
   .ADDRA(ADDRA),   // Input port-A address, width defined by Port A depth
   .ADDRB(ADDRB),   // Input port-B address, width defined by Port B depth
   .CLKA(CLKA),     // 1-bit input port-A clock
   .CLKB(CLKB),     // 1-bit input port-B clock
   .DIA(DIA),       // Input port-A data, width defined by WRITE_WIDTH_A parameter
   .DIB(DIB),       // Input port-B data, width defined by WRITE_WIDTH_B parameter
   .ENA(ENA),       // 1-bit input port-A enable
   .ENB(ENB),       // 1-bit input port-B enable
   .REGCEA(REGCEA), // 1-bit input port-A output register enable
   .REGCEB(REGCEB), // 1-bit input port-B output register enable
   .RSTA(RSTA),     // 1-bit input port-A reset
   .RSTB(RSTB),     // 1-bit input port-B reset
   .WEA(WEA),       // Input port-A write enable, width defined by Port A depth
   .WEB(WEB)        // Input port-B write enable, width defined by Port B depth
);

// End of BRAM_TDP_MACRO_inst instantiation
```

For More Information

See the *Spartan-6 FPGA User Documentation (User Guides and Data Sheets)*.

ADDMACC_MACRO

Macro: Adder/Multiplier/Accumulator



Introduction

The ADDMACC _MACRO simplifies the instantiation of the DSP48 block when used as a pre-add, multiply accumulate function. It features parameterizable input and output widths and latency that ease the integration of DSP48 block into HDL.

Port Description

Name	Direction	Width	Function
Output Ports			
PRODUCT	Output	Variable width, equals the value of the WIDTH_A attibute plus the value of the WIDTH_B attribute.	Primary data output.
Input Ports			
PREADD1	Input	Variable, see WIDTH_PREADD attribute.	Preadder data input.
PREADD2	Input	Variable, see WIDTH_PREADD attribute.	Preadder data input
MULTIPLIER	Input	Variable, see WIDTH_MULTIPLIER attribute.	Multiplier data input
CARRYIN	Input	1	Carry input
CLK	Input	1	Clock
CE	Inupt	1	Clock enable
LOAD	Input	1	Load
LOAD_DATA	Input	Variable, see WIDTH_PRODUCT attribute.	In a DSP slice, when LOAD is asserted, loads P with A*B+LOAD_DATA.

Name	Direction	Width	Function
RST	Input	1	Synchronous Reset

Design Entry Method

This unimacro can be instantiated only. It is a parameterizable version of the primitive.

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

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
WIDTH_PREADD	Integer	1 to 24	24	Controls the width of PREADD1 and PREADD2 inputs.
WIDTH_MULTIPLIER	Integer	1 to 18	18	Controls the width of MULTIPLIER input.
WIDTH_PRODUCT	Integer	1 to 48	48	Controls the width of MULTIPLIER output.
LATENCY	Integer	0, 1, 2, 3, 4	3	Number of pipeline registers • 1 - MREG == 1 • 2 - AREG == BREG == 1 and MREG == 1 or MREG == 1 and PREG == 1 • 3 - AREG == BREG == 1 and MREG == 1 and PREG == 1 • 4 - AREG == BREG == 2 and MREG == 1 and PREG == 1
DEVICE	String	"VIRTEX6", "SPARTAN6"	"VIRTEX6"	Target hardware architecture.

VHDL Instantiation Template

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

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


-- ADDMACC_MACRO: Add and Multiple Accumulate Function implemented in a DSP48E
--                Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

ADDMACC_MACRO_inst : ADDMACC_MACRO
generic map (
   DEVICE => "SPARTAN6",  -- Target Device: "VIRTEX6", "SPARTAN6"
   LATENCY => 4,          -- Desired clock cycle latency, 1-4
   WIDTH_PREADD => 25,    -- Pre-Adder input bus width, 1-25
   WIDTH_MULTIPLIER => 18, -- Multiplier input bus width, 1-18
   WIDTH_PRODUCT => 48)    -- MACC output width, 1-48
port map (
   PRODUCT => PRODUCT,     -- MACC result output, width defined by WIDTH_PRODUCT generic
   MULTIPLIER => MULTIPLIER, -- Multiplier data input, width determined by WIDTH_MULTIPLIER generic
```

```
   PREADDER1 => PREADDER1,   -- Preadder data input, width determined by WIDTH_PREADDER generic
   PREADDER2 => PREADDER2,   -- Preadder data input, width determined by WIDTH_PREADDER generic
   CARRYIN => CARRYIN, -- 1-bit carry-in input
   CE => CE,       -- 1-bit input clock enable
   CLK => CLK,     -- 1-bit clock input
   LOAD => LOAD, -- 1-bit accumulator load input
   LOAD_DATA => LOAD_DATA, -- Accumulator load data input, width defined by WIDTH_PRODUCT generic
   RST => RST    -- 1-bit input active high synchronous reset
);
-- End of ADDMACC_MACRO_inst instantiation
```

Verilog Instantiation Template

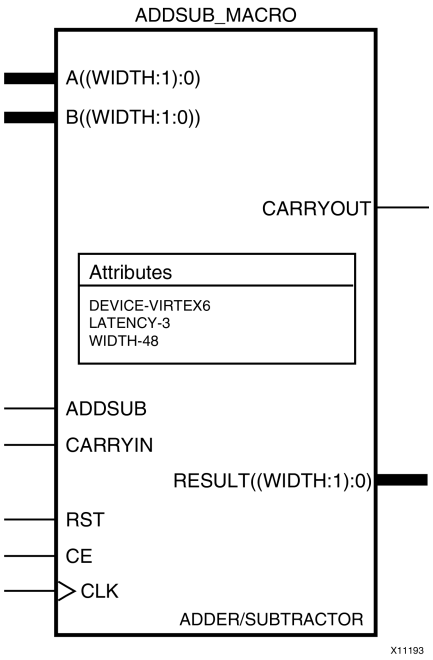
```
// ADDMACC_MACRO: Variable width & latency - Pre-Add -> Multiplier -> Accumulate
//                function implemented in a DSP48E
//                Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

ADDMACC_MACRO #(
   .DEVICE("SPARTAN6"),   // Target Device: "VIRTEX6", "SPARTAN6"
   .LATENCY(4),           // Desired clock cycle latency, 0-4
   .WIDTH_PREADD(18),     // Pre-adder input width, 1-18
   .WIDTH_MULTIPLIER(18), // Multiplier input width, 1-18
   .WIDTH_PRODUCT(48)     // MACC output width, 1-48
) ADDMACC_MACRO_inst (
   .PRODUCT(PRODUCT),   // MACC result output, width defined by WIDTH_PRODUCT parameter
   .CARRYIN(CARRYIN),   // 1-bit carry-in input
   .CLK(CLK),           // 1-bit clock input
   .CE(CE),             // 1-bit clock enable input
   .LOAD(LOAD),         // 1-bit accumulator load input
   .LOAD_DATA(LOAD_DATA),   // Accumulator load data input, width defined by WIDTH_PRODUCT parameter
   .MULTIPLIER(MULTIPLIER), // Multiplier data input, width defined by WIDTH_MULTIPLIER parameter
   .PREADD2(PREADD2),   // Preadder data input, width defined by WIDTH_PREADD parameter
   .PREADD1(PREADD1),   // Preadder data input, width defined by WIDTH_PREADD parameter
   .RST(RST)            // 1-bit active high synchronous reset
);

// End of ADDMACC_MACRO_inst instantiation
```

For More Information

See the *Spartan-6 FPGA User Documentation (User Guides and Data Sheets)*.

ADDSUB_MACRO

Macro: Adder/Subtractor



Introduction

The ADDSUB_MACRO simplifies the instantiation of the DSP48 block when used as a simple adder/subtractor. It features parameterizable input and output widths and latency that ease the integration of the DSP48 block into HDL.

Port Description

Name	Direction	Width (Bits)	Function
Output Ports			
CARRYOUT	Output	1	Carry Out
RESULT	Output	Variable, see WIDTH attrribute.	Data output bus addressed by RDADDR.
Input Ports			
ADDSUB	Input	1	When high, RESULT is an addition. When low, RESULT is a subtraction.
A	Input	Variable, see WIDTH attribute.	Data input to add/sub.
B	Input	Variable, see WIDTH attribute.	Data input to add/sub
CE	Input	1	Clock Enable
CARRYIN	Input	1	Carry In
CLK	Input	1	Clock
RST	Input	1	Synchronous Reset

Design Entry Method

This unimacro can be instantiated only. It is a parameterizable version of the primitive.

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

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
DEVICE	String	"VIRTEX6", "SPARTAN6"	"VIRTEX6"	Target hardware architecture.
LATENCY	Integer	0, 1, 2	2	Number of pipeline registers. • 1 - PREG == 1 • 2 - AREG == BREG == CREG == PREG
WIDTH	Integer	1-48	48	A, B, and RESULT port width; internal customers can override B and RESULT port widths using other parameters
WIDTH_RESULT	Integer	1-48	48	Result port width override.

VHDL Instantiation Template

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

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


-- ADDSUB_MACRO: Variable width & latency - Adder / Subtrator implemented in a DSP48E
--               Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

ADDSUB_MACRO_inst : ADDSUB_MACRO
generic map (
   DEVICE => "SPARTAN6", -- Target Device: "VIRTEX5", "VIRTEX6", "SPARTAN6"
   LATENCY => 2,         -- Desired clock cycle latency, 0-2
   WIDTH => 48)          -- Input / Output bus width, 1-48
port map (
   CARRYOUT => CARRYOUT, -- 1-bit carry-out output signal
   RESULT => RESULT,     -- Add/sub result output, width defined by WIDTH generic
   A => A,               -- Input A bus, width defined by WIDTH generic
   ADD_SUB => ADD_SUB,   -- 1-bit add/sub input, high selects add, low selects subtract
   B => B,               -- Input B bus, width defined by WIDTH generic
   CARRYIN => CARRYIN,   -- 1-bit carry-in input
   CE => CE,             -- 1-bit clock enable input
   CLK =>CLK,           -- 1-bit clock input
   RST => RST            -- 1-bit active high synchronous reset
);
-- End of ADDSUB_MACRO_inst instantiation
```

Verilog Instantiation Template

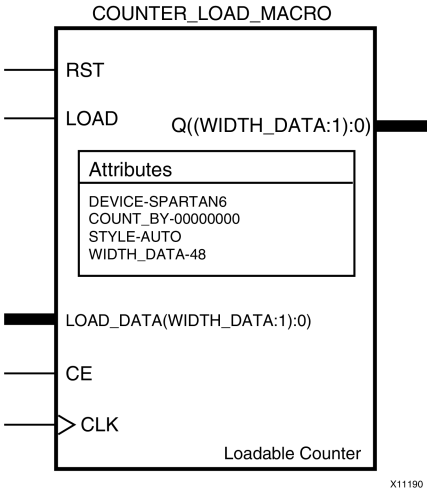
```
// ADDSUB_MACRO: Variable width & latency - Adder / Subtrator implemented in a DSP48E
//               Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

ADDSUB_MACRO #(
   .DEVICE("SPARTAN6"), // Target Device: "VIRTEX5", "VIRTEX6", "SPARTAN6"
   .LATENCY(2),        // Desired clock cycle latency, 0-2
   .WIDTH(48)          // Input / output bus width, 1-48
) ADDSUB_MACRO_inst (
   .CARRYOUT(CARRYOUT), // 1-bit carry-out output signal
   .RESULT(RESULT),     // Add/sub result output, width defined by WIDTH parameter
   .A(A),               // Input A bus, width defined by WIDTH parameter
   .ADD_SUB(ADD_SUB),   // 1-bit add/sub input, high selects add, low selects subtract
   .B(B),               // Input B bus, width defined by WIDTH parameter
   .CARRYIN(CARRYIN),   // 1-bit carry-in input
   .CE(CE),             // 1-bit clock enable input
   .CLK(CLK),           // 1-bit clock input
   .RST(RST)            // 1-bit active high synchronous reset
);

// End of ADDSUB_MACRO_inst instantiation
```

For More Information

See the *Spartan-6 FPGA User Documentation (User Guides and Data Sheets)*.

COUNTER_LOAD_MACRO

Macro: Loadable Counter



Introduction

The COUNTER_LOAD_MACRO simplifies the instantiation of the DSP48 block when used as dynamic loading up/down counter. It features parameterizable output width and count by values that ease the integration of the DSP48 block into HDL.

Port Description

Name	Direction	Width	Function
Output Ports			
Q	Output	Variable, see WIDTH_DATA attribute.	Counter output.
Input Ports			
CE	Input	1	Clock Enable.
CLK	Input	1	Clock.
LOAD	Input	Variable, see WIDTH_DATA attribute.	When asserted, loads the counter from LOAD_DATA (two-clock latency).
LOAD_DATA	Input	Variable, see WIDTH_DATA attribute.	In a DSP slice, asserting the LOAD pin will force this data into the P register with a latency of 2 clocks.
DIRECTION	Input	1	High for Up and Low for Down (two-clock latency)
RST	Input	1	Synchronous Reset

Design Entry Method

This unimacro can be instantiated only. It is a parameterizable version of the primitive.

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

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
DEVICE	String	"VIRTEX6", "SPARTAN6"	"VIRTEX6"	Target hardware architecture.
COUNT_BY	Hexa-decimal	Any 48 bit value.	000000000001	Count by *n*; takes precedence over WIDTH_DATA.
WIDTH_DATA	Integer	1-48	48	Specifies counter width.

VHDL Instantiation Template

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

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


-- COUNTER_LOAD_MACRO: Loadable variable counter implemented in a DSP48E
--                     Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

COUNTER_LOAD_MACRO_inst : COUNTER_LOAD_MACRO
generic map (
   COUNT_BY => X"000000000001", -- Count by value
   DEVICE => "SPARTAN6",         -- Target Device: "VIRTEX5", "VIRTEX6", "SPARTAN6"
   WIDTH_DATA => 48)            -- Counter output bus width, 1-48
port map (
   Q => Q,                 -- Counter ouput, width determined by WIDTH_DATA generic
   CLK => CLK,             -- 1-bit clock input
   CE => CE,               -- 1-bit clock enable input
   DIRECTION => DIRECTION, -- 1-bit up/down count direction input, high is count up
   LOAD => LOAD,           -- 1-bit active high load input
   LOAD_DATA => LOAD_DATA, -- Counter load data, width determined by WIDTH_DATA generic
   RST => RST              -- 1-bit active high synchronous reset
);
-- End of COUNTER_LOAD_MACRO_inst instantiation
```

Verilog Instantiation Template

```
// COUNTER_LOAD_MACRO: Loadable variable counter implemented in a DSP48E
//                     Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

COUNTER_LOAD_MACRO #(
   .COUNT_BY(48'h000000000001), // Count by value
   .DEVICE("SPARTAN6"), // Target Device: "VIRTEX5", "VIRTEX6", "SPARTAN6"
   .WIDTH_DATA(48)     // Counter output bus width, 1-48
) COUNTER_LOAD_MACRO_inst (
   .Q(Q),                  // Counter output, width determined by WIDTH_DATA parameter
   .CLK(CLK),              // 1-bit clock input
   .CE(CE),                // 1-bit clock enable input
   .DIRECTION(DIRECTION), // 1-bit up/down count direction input, high is count up
   .LOAD(LOAD),            // 1-bit active high load input
   .LOAD_DATA(LOAD_DATA), // Counter load data, width determined by WIDTH_DATA parameter
   .RST(RST)               // 1-bit active high synchronous reset
);
```

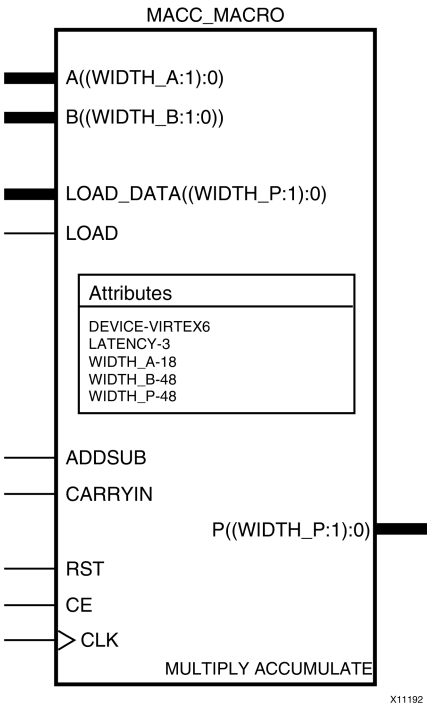
```
// End of COUNTER_LOAD_MACRO_inst instantiation
```

For More Information

See the *Spartan-6 FPGA User Documentation (User Guides and Data Sheets)*.

MACC_MACRO

Macro: Multiplier/Accumulator



Introduction

The MACC_MACRO simplifies the instantiation of the DSP48 block when used in simple signed multiplier/accumulator mode. It features parameterizable input and output widths and latencies that ease the integration of the DSP48 block into HDL.

Port Description

Name	Direction	Width	Function
Output Ports			
P	Output	Variable width, equals the value of the WIDTH_A attibute plus the value of the WIDTH_B attribute.	Primary data output.
Input Ports			
A	Input	Variable, see WIDTH_A attribute.	Multiplier data input.
B	Input	Variable, see WIDTH_B attribute.	Multiplier data input.
CARRYIN	Input	1	Carry input.
CE	Input	1	Clock enable.
CLK	Input	1	Clock.
LOAD	Inupt	1	Load.
LOAD_DATA	Input	Variable width, equals the value of the WIDTH_A attibute plus the value of the WIDTH_B attribute.	In a DSP slice, when LOAD is asserted, loads P with A*B+LOAD_DATA.

Name	Direction	Width	Function
RST	Input	1	Synchronous Reset.
ADDSUB	Input	1	High sets accumulator in addition mode; low sets accumulator in subtraction mode.

Design Entry Method

This unimacro can be instantiated only. It is a parameterizable version of the primitive.

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

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
WIDTH_A	Integer	1 to 18	18	Controls the width of A input.
WIDTH_B	Integer	1 to 18	18	Controls the width of B input
LATENCY	Integer	0, 1, 2, 3, 4	3	Number of pipeline registers • 1 - MREG == 1 • 2 - AREG == BREG == 1 and MREG == 1 or MREG == 1 and PREG == 1 • 3 - AREG == BREG == 1 and MREG == 1 and PREG == 1 • 4 - AREG == BREG == 2 and MREG == 1 and PREG == 1
DEVICE	String	"VIRTEX5", "VIRTEX6", "SPARTAN6"	"VIRTEX6"	Target hardware architecture.

VHDL Instantiation Template

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

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


-- MACC_MACRO: Multiple Accumulate Function implemented in a DSP48E
--             Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

MACC_MACRO_inst : MACC_MACRO
generic map (
   DEVICE => "SPARTAN6",  -- Target Device: "VIRTEX5", "VIRTEX6", "SPARTAN6"
   LATENCY => 3,          -- Desired clock cycle latency, 1-4
   WIDTH_A => 25,         -- Multiplier A-input bus width, 1-25
   WIDTH_B => 18,         -- Multiplier B-input bus width, 1-18
   WIDTH_P => 48)         -- Accumulator output bus width, 1-48
port map (
   P => P,     -- MACC ouput bus, width determined by WIDTH_P generic
   A => A,     -- MACC input A bus, width determined by WIDTH_A generic
   ADDSUB => ADDSUB, -- 1-bit add/sub input, high selects add, low selects subtract
   B => B,           -- MACC input B bus, width determined by WIDTH_B generic
```

```
   CARRYIN => CARRYIN, -- 1-bit carry-in input to accumulator
   CE => CE,       -- 1-bit active high input clock enable
   CLK => CLK,     -- 1-bit positive edge clock input
   LOAD => LOAD, -- 1-bit active high input load accumulator enable
   LOAD_DATA => LOAD_DATA, -- Load accumulator input data,
                           -- width determined by WIDTH_P generic
   RST => RST    -- 1-bit input active high reset
);

-- End of MACC_MACRO_inst instantiation
```

Verilog Instantiation Template

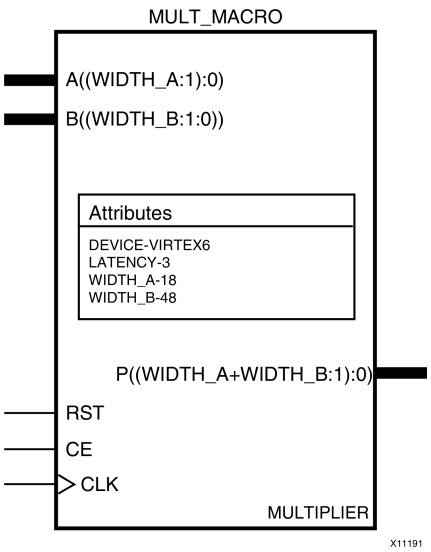
```
// MACC_MACRO: Multiply Accumulate Function implemented in a DSP48E
//             Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

MACC_MACRO #(
   .DEVICE("SPARTAN6"), // Target Device: "VIRTEX5", "VIRTEX6", "SPARTAN6"
   .LATENCY(3),        // Desired clock cycle latency, 1-4
   .WIDTH_A(18),       // Multiplier A-input bus width, 1-18
   .WIDTH_B(18),       // Multiplier B-input bus width, 1-18
   .WIDTH_P(48)        // Accumulator output bus width, 1-48
) MACC_MACRO_inst (
   .P(P),     // MACC output bus, width determined by WIDTH_P parameter
   .A(A),     // MACC input A bus, width determined by WIDTH_A parameter
   .ADDSUB(ADDSUB), // 1-bit add/sub input, high selects add, low selects subtract
   .B(B),     // MACC input B bus, width determined by WIDTH_B parameter
   .CARRYIN(CARRYIN), // 1-bit carry-in input to accumulator
   .CE(CE),     // 1-bit active high input clock enable
   .CLK(CLK),   // 1-bit positive edge clock input
   .LOAD(LOAD), // 1-bit active high input load accumulator enable
   .LOAD_DATA(LOAD_DATA), // Load accumulator input data, width determined by WIDTH_P parameter
   .RST(RST)    // 1-bit input active high reset
);

// End of MACC_MACRO_inst instantiation
```

For More Information

See the *Spartan-6 FPGA User Documentation (User Guides and Data Sheets)*.

MULT_MACRO

Macro: Multiplier



Introduction

The MULT_MACRO simplifies the instantiation of the DSP48 block when used as a simple signed multiplier. It features parameterizable input and output widths and latencies that ease the integration of the DSP48 block into HDL.

Port Description

Name	Direction	Width	Function
Output Ports			
P	Output	Variable width, equals the value of the WIDTH_A attibute plus the value of the WIDTH_B attribute.	Primary data output.
Input Ports			
A	Input	Variable, see WIDTH_A attribute.	Multiplier data input.
B	Input	Variable, see WIDTH_B attribute.	Multiplier data input.
CE	Input	1	Clock Enable.
CLK	Input	1	Clock.
RST	Input	1	Synchronous Reset.

Design Entry Method

This unimacro can be instantiated only. It is a parameterizable version of the primitive.

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

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
WIDTH_A	Integer	1 to 18	18	Controls the width of A input.
WIDTH_B	Integer	1 to 18	18	Controls the width of B input.
LATENCY	Integer	0, 1, 2, 3, 4	3	Number of pipeline registers • 1 - MREG == 1 • 2 - AREG == BREG == 1 and MREG == 1 or MREG == 1 and PREG == 1 • 3 - AREG == BREG == 1 and MREG == 1 and PREG == 1 • 4 - AREG == BREG == 2 and MREG == 1 and PREG == 1
DEVICE	String	"VIRTEX5", "VIRTEX6", "SPARTAN6"	VIRTEX6"	Target hardware architecture.

VHDL 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_MACRO: Multiply Function implemented in a DSP48E
--             Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

MULT_MACRO_inst : MULT_MACRO
generic map (
   DEVICE => "SPARTAN6",    -- Target Device: "VIRTEX5", "VIRTEX6", "SPARTAN6"
   LATENCY => 3,           -- Desired clock cycle latency, 0-4
   WIDTH_A => 18,          -- Multiplier A-input bus width, 1-25
   WIDTH_B => 18)          -- Multiplier B-input bus width, 1-18
port map (
   P => P,     -- Multiplier ouput bus, width determined by WIDTH_P generic
   A => A,     -- Multiplier input A bus, width determined by WIDTH_A generic
   B => B,     -- Multiplier input B bus, width determined by WIDTH_B generic
   CE => CE,   -- 1-bit active high input clock enable
   CLK => CLK, -- 1-bit positive edge clock input
   RST => RST  -- 1-bit input active high reset
);
-- End of MULT_MACRO_inst instantiation
```

Verilog Instantiation Template

```
// MULT_MACRO: Multiply Function implemented in a DSP48E
//             Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

MULT_MACRO #(
   .DEVICE("SPARTAN6"), // Target Device: "VIRTEX5", "VIRTEX6", "SPARTAN6"
   .LATENCY(3),        // Desired clock cycle latency, 0-4
   .WIDTH_A(18),       // Multiplier A-input bus width, 1-18
   .WIDTH_B(18)        // Multiplier B-input bus width, 1-18
) MULT_MACRO_inst (
   .P(P),     // Multiplier output bus, width determined by WIDTH_P parameter
   .A(A),     // Multiplier input A bus, width determined by WIDTH_A parameter
   .B(B),     // Multiplier input B bus, width determined by WIDTH_B parameter
   .CE(CE),   // 1-bit active high input clock enable
   .CLK(CLK), // 1-bit positive edge clock input
```

```
   .RST(RST)  // 1-bit input active high reset
);

// End of MULT_MACRO_inst instantiation
```

For More Information

See the *Spartan-6 FPGA User Documentation (User Guides and Data Sheets)*.

XILINX®

Chapter 3

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.

Advanced

Design Element	Description
MCB	Primitive: Memory Control Block
PCIE_A1	Primitive: PCI Express

Arithmetic Functions

Design Element	Description
DSP48A1	Primitive: Multi-Functional, Cascadable, 48-bit Output, Arithmetic Block

Clock Components

Design Element	Description
BUFG	Convenience Primitive: Global Clock Buffer
BUFGCE	Convenience Primitive: Global Clock Buffer with Clock Enable
BUFGMUX	Primitive: Global Clock MUX Buffer
BUFGMUX_1	Primitive: Global Clock MUX Buffer with Output State 1
BUFH	Primitive: Clock buffer for a single clocking region
BUFIO2	Primitive: Dual Clock Buffer and Strobe Pulse
BUFIO2_2CLK	Primitive: Dual Clock Buffer and Strobe Pulse with Differential Input

Design Element	Description
BUFIO2FB	Primitive: Feedback Clock Buffer
BUFPLL	Primitive: PLL Buffer
BUFPLL_MCB	Primitive: PLL Buffer for the Memory Controller Block
DCM_CLKGEN	Primitive: Digital Clock Manager.
DCM_SP	Primitive: Digital Clock Manager
PLL_BASE	Primitive: Basic Phase Locked Loop Clock Circuit

Config/BSCAN Components

Design Element	Description
BSCAN_SPARTAN6	Primitive: Spartan®-6 JTAG Boundary Scan Logic Control Circuit
DNA_PORT	Primitive: Device DNA Data Access Port
ICAP_SPARTAN6	Primitive: Internal Configuration Access Port
JTAG_SIM_SPARTAN6	Simulation: JTAG TAP Controller Simulation Model
POST_CRC_INTERNAL	Primitive: Post-configuration CRC error detection
SIM_CONFIG_S6	Simulation: Configuration Simulation Model
SIM_CONFIG_S6_SERIAL	Simulation: Serial Configuration Simulation Model
STARTUP_SPARTAN6	Primitive: Spartan®-6 Global Set/Reset, Global 3-State and Configuration Start-Up Clock Interface
SUSPEND_SYNC	Primitive: Suspend Mode Access

Convenience Primitives

Design Element	Description
BUFGCE_1	Convenience Primitive: Global Clock Buffer with Clock Enable and Output State 1

I/O Components

Design Element	Description
GTPA1_DUAL	Primitive: Dual Gigabit Transceiver
IBUF	Primitive: Input Buffer
IBUFDS	Primitive: Differential Signaling Input Buffer
IBUFDS_DIFF_OUT	Primitive: Signaling Input Buffer with Differential Output
IBUFG	Primitive: Dedicated Input Clock Buffer
IBUFGDS	Primitive: Differential Signaling Dedicated Input Clock Buffer and Optional Delay
IBUFGDS_DIFF_OUT	Primitive: Differential Signaling Input Buffer with Differential Output
IOBUF	Primitive: Bi-Directional Buffer
IOBUFDS	Primitive: 3-State Differential Signaling I/O Buffer with Active Low Output Enable

Design Element	Description
IODELAY2	Primitive: Input and Output Fixed or Variable Delay Element
IODRP2	Primitive: I/O Control Port
IODRP2_MCB	Primitive: I/O Control Port for the Memory Controller Block
ISERDES2	Primitive: Input SERial/DESerializer.
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
OSERDES2	Primitive: Dedicated IOB Output Serializer
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
RAM128X1D	Primitive: 128-Deep by 1-Wide Dual Port Random Access Memory (Select RAM)
RAM256X1S	Primitive: 256-Deep by 1-Wide Random Access Memory (Select RAM)
RAM32M	Primitive: 32-Deep by 8-bit Wide Multi Port Random Access Memory (Select RAM)
RAM32X1D	Primitive: 32-Deep by 1-Wide Static Dual Port Synchronous RAM
RAM32X1S	Primitive: 32-Deep by 1-Wide Static Synchronous RAM
RAM32X1S_1	Primitive: 32-Deep by 1-Wide Static Synchronous RAM with Negative-Edge Clock
RAM32X2S	Primitive: 32-Deep by 2-Wide Static Synchronous RAM
RAM64M	Primitive: 64-Deep by 4-bit Wide Multi Port Random Access Memory (Select RAM)
RAM64X1D	Primitive: 64-Deep by 1-Wide Dual Port Static Synchronous RAM
RAM64X1S	Primitive: 64-Deep by 1-Wide Static Synchronous RAM
RAM64X1S_1	Primitive: 64-Deep by 1-Wide Static Synchronous RAM with Negative-Edge Clock
RAMB16BWER	Primitive: 16K-bit Data and 2K-bit Parity Configurable Synchronous Dual Port Block RAM with Optional Output Registers

Design Element	Description
RAMB8BWER	Primitive: 8K-bit Data and 1K-bit Parity Configurable Synchronous Dual Port Block RAM with Optional Output Registers
ROM128X1	Primitive: 128-Deep by 1-Wide ROM
ROM256X1	Primitive: 256-Deep by 1-Wide ROM
ROM32X1	Primitive: 32-Deep by 1-Wide ROM
ROM64X1	Primitive: 64-Deep by 1-Wide ROM

Registers/Latches

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

Shift Register LUT

Design Element	Description
SRL16E	Primitive: 16-Bit Shift Register Look-Up Table (LUT) with Clock Enable
SRLC32E	Primitive: 32 Clock Cycle, Variable Length Shift Register Look-Up Table (LUT) with Clock Enable

Slice/CLB Primitives

Design Element	Description
CARRY4	Primitive: Fast Carry Logic with Look Ahead
CFGLUT5	Primitive: 5-input Dynamically Reconfigurable Look-Up Table (LUT)
LUT1	Macro: 1-Bit Look-Up Table with General Output
LUT1_D	Macro: 1-Bit Look-Up Table with Dual Output
LUT1_L	Macro: 1-Bit Look-Up Table with Local Output
LUT2	Macro: 2-Bit Look-Up Table with General Output
LUT2_D	Macro: 2-Bit Look-Up Table with Dual Output
LUT2_L	Macro: 2-Bit Look-Up Table with Local Output
LUT3	Macro: 3-Bit Look-Up Table with General Output
LUT3_D	Macro: 3-Bit Look-Up Table with Dual Output
LUT3_L	Macro: 3-Bit Look-Up Table with Local Output
LUT4	Macro: 4-Bit Look-Up-Table with General Output
LUT4_D	Macro: 4-Bit Look-Up Table with Dual Output
LUT4_L	Macro: 4-Bit Look-Up Table with Local Output
LUT5	Primitive: 5-Input Lookup Table with General Output
LUT5_D	Primitive: 5-Input Lookup Table with General and Local Outputs
LUT5_L	Primitive: 5-Input Lookup Table with Local Output
LUT6	Primitive: 6-Input Lookup Table with General Output
LUT6_2	Primitive: Six-input, 2-output, Look-Up Table
LUT6_D	Primitive: 6-Input Lookup Table with General and Local Outputs
LUT6_L	Primitive: 6-Input Lookup Table 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

XILINX®

Chapter 4

About Design Elements

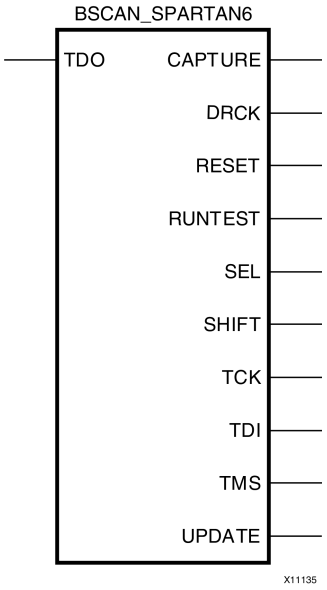
This section describes the design elements that can be used with this architecture. The design elements are organized alphabetically.

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

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

BSCAN_SPARTAN6

Primitive: Spartan®-6 JTAG Boundary Scan Logic Control Circuit



Introduction

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

Each instance of this design element will handle one JTAG USER instruction (USER1 through USER4) as set with the JTAG_CHAIN attribute. To handle all four USER instuctions, instantiate four of these elements and set the JTAG_CHAIN attribute appropriately.

Note For specific information on boundary scan for an architecture, see the *Spartan-6 Configuration User Guide* for this element.

Port Descriptions

Port	Direction	Width	Function
CAPTURE	Output	1	CAPTURE output from TAP controller.
DRCK	Output	1	Data register output for USER functions.
RESET	Output	1	Reset output for TAP controller.
RUNTEST	Output	1	Output signal that gets asserted when TAP controller is in Run Test Idle state.
SEL	Output	1	USER active output.
SHIFT	Output	1	SHIFT output from TAP controller.
TCK	Output	1	Scan Clock output. Fabric connection to TAP Clock pin.
TDI	Output	1	TDI output from TAP controller.
TDO	Input	1	Data input for USER function.
TMS	Output	1	Test Mode Select output. Fabric connection to TAP.
UPDATE	Output	1	UPDATE output from TAP controller.

Design Entry Method

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

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
JTAG_CHAIN	Integer	1, 2, 3, 4	1	Sets the JTAG USER instruction number that this instance of the element will handle.

VHDL 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_SPARTAN6: JTAG Boundary Scan Logic Control Circuit
--                 Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

BSCAN_SPARTAN6_inst : BSCAN_SPARTAN6
generic map (
   JTAG_CHAIN => 1  -- Value for USER command. Possible values: (1,2,3 or 4).
)
port map (
   CAPTURE => CAPTURE, -- 1-bit output: CAPTURE output from TAP controller.
   DRCK => DRCK,       -- 1-bit output: Data register output for USER functions.
   RESET => RESET,     -- 1-bit output: Reset output for TAP controller.
   RUNTEST => RUNTEST, -- 1-bit output: Output signal that gets asserted when TAP controller is in Run Test
                       -- Idle state.

   SEL => SEL,         -- 1-bit output: USER active output.
   SHIFT => SHIFT,     -- 1-bit output: SHIFT output from TAP controller.
   TCK => TCK,         -- 1-bit output: Scan Clock output. Fabric connection to TAP Clock pin.
   TDI => TDI,         -- 1-bit output: TDI output from TAP controller.
   TMS => TMS,         -- 1-bit output: Test Mode Select output. Fabric connection to TAP.
   UPDATE => UPDATE,   -- 1-bit output: UPDATE output from TAP controller
   TDO => TDO          -- 1-bit input: Data input for USER function.
);

-- End of BSCAN_SPARTAN6_inst instantiation
```

Verilog Instantiation Template

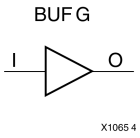
```
// BSCAN_SPARTAN6: JTAG Boundary Scan Logic Control Circuit
//                 Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

BSCAN_SPARTAN6 #(
   .JTAG_CHAIN(1)  // Value for USER command. Possible values: (1,2,3 or 4).
)
BSCAN_SPARTAN6_inst (
   .CAPTURE(CAPTURE), // 1-bit output: CAPTURE output from TAP controller.
   .DRCK(DRCK),       // 1-bit output: Data register output for USER functions.
   .RESET(RESET),     // 1-bit output: Reset output for TAP controller.
   .RUNTEST(RUNTEST), // 1-bit output: Output signal that gets asserted when TAP controller is in Run Test
                      // Idle state.

   .SEL(SEL),         // 1-bit output: USER active output.
   .SHIFT(SHIFT),     // 1-bit output: SHIFT output from TAP controller.
   .TCK(TCK),         // 1-bit output: Scan Clock output. Fabric connection to TAP Clock pin.
   .TDI(TDI),         // 1-bit output: TDI output from TAP controller.
   .TMS(TMS),         // 1-bit output: Test Mode Select output. Fabric connection to TAP.
   .UPDATE(UPDATE),   // 1-bit output: UPDATE output from TAP controller
   .TDO(TDO)          // 1-bit input: Data input for USER function.
);

// End of BSCAN_SPARTAN6_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA Configuration User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

BUFG

Primitive: Global Clock Buffer



Introduction

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

Port Descriptions

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

Design Entry Method

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

VHDL Instantiation Template

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

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


-- BUFG: Global Clock Buffer
--       Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of BUFG_inst instantiation
```

Verilog Instantiation Template

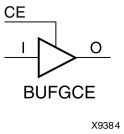
```
// BUFG: Global Clock Buffer
//       Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

BUFG BUFG_inst (
   .O(O), // 1-bit output: Clock buffer output
   .I(I)  // 1-bit input: Clock buffer input
);

// End of BUFG_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA Clocking Resources User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

BUFGCE

Primitive: Global Clock Buffer with Clock Enable



Introduction

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

Logic Table

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

Port Descriptions

Port	Direction	Width	Function
I	Input	1	Clock buffer input
CE	Input	1	Clock enable input
O	Output	1	Clock buffer output

Design Entry Method

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

VHDL Instantiation Template

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

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


-- BUFGCE: Global Clock Buffer with Clock Enable
--         Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

BUFGCE_inst : BUFGCE
port map (
   O => O,   -- 1-bit output: Clock buffer output
   CE => CE, -- 1-bit input: Clock buffer select
   I => I    -- 1-bit input: Clock buffer input (S=0)
);

-- End of BUFGCE_inst instantiation
```

Verilog Instantiation Template

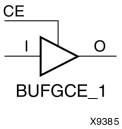
```
// BUFGCE: Global Clock Buffer with Clock Enable
//         Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

BUFGCE BUFGCE_inst (
   .O(O),   // 1-bit output: Clock buffer output
   .CE(CE), // 1-bit input: Clock buffer select
   .I(I)    // 1-bit input: Clock buffer input (S=0)
);

// End of BUFGCE_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA Clocking Resources User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

BUFGCE_1

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



Introduction

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

Logic Table

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

Port Descriptions

Port	Direction	Width	Function
I	Input	1	Clock buffer input
CE	Input	1	Clock enable input
O	Output	1	Clock buffer output

Design Entry Method

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

VHDL Instantiation Template

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

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


-- BUFGCE_1: Global Clock Buffer with Clock Enable and Output State 1
--           Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

BUFGCE_1_inst : BUFGCE_1
port map (
   O => O,   -- 1-bit output: Clock buffer output
   CE => CE, -- 1-bit input: Clock buffer select
   I => I    -- 1-bit input: Clock buffer input (S=0)
);
```

```
-- End of BUFGCE_1_inst instantiation
```

Verilog Instantiation Template

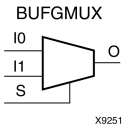
```
// BUFGCE_1: Global Clock Buffer with Clock Enable and Output State 1
//           Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

BUFGCE_1 BUFGCE_1_inst (
   .O(O),   // 1-bit output: Clock buffer output
   .CE(CE), // 1-bit input: Clock buffer select
   .I(I)    // 1-bit input: Clock buffer input (S=0)
);

// End of BUFGCE_1_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA Clocking Resources User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

BUFGMUX

Primitive: Global Clock MUX Buffer



Introduction

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

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

Port Descriptions

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

Design Entry Method

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

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
CLK_SEL_TYPE	String	"SYNC", "ASYNC"	"SYNC"	Specifies synchronous or asynchronous clock.

VHDL 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 Mux Buffer
--          Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

BUFGMUX_inst : BUFGMUX
generic map (
   CLK_SEL_TYPE => "SYNC"  -- Glitchles ("SYNC") or fast ("ASYNC") clock switch-over
)
port map (
   O => O,   -- 1-bit output: Clock buffer output
   I0 => I0, -- 1-bit input: Clock buffer input (S=0)
   I1 => I1, -- 1-bit input: Clock buffer input (S=1)
   S => S    -- 1-bit input: Clock buffer select
);

-- End of BUFGMUX_inst instantiation
```

Verilog Instantiation Template

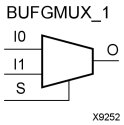
```
// BUFGMUX: Global Clock Mux Buffer
//          Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

BUFGMUX #(
   .CLK_SEL_TYPE("SYNC")  // Glitchles ("SYNC") or fast ("ASYNC") clock switch-over
)
BUFGMUX_inst (
   .O(O),   // 1-bit output: Clock buffer output
   .I0(I0), // 1-bit input: Clock buffer input (S=0)
   .I1(I1), // 1-bit input: Clock buffer input (S=1)
   .S(S)    // 1-bit input: Clock buffer select
);

// End of BUFGMUX_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA Clocking Resources User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

BUFGMUX_1

Primitive: Global Clock MUX Buffer with Output State 1



Introduction

This design element is a multiplexed global clock buffer that can select between two input clocks: I0 and I1. When the select input (S) is Low, the signal on I0 is selected for output (0). When the select input (S) is High, the signal on I1 is selected for output.

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

Logic Table

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

Port Descriptions

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

Design Entry Method

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

VHDL Instantiation Template

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

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


-- BUFGMUX_1: Global Clock Mux Buffer with Output State 1
--            Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

BUFGMUX_1_inst : BUFGMUX_1
generic map (
   CLK_SEL_TYPE => "SYNC"  -- Glitchles ("SYNC") or fast ("ASYNC") clock switch-over
)
port map (
   O => O,   -- 1-bit output: Clock buffer output
   I0 => I0, -- 1-bit input: Clock buffer input
   I1 => I1, -- 1-bit input: Clock buffer input
   S => S    -- 1-bit input: Clock buffer select
);

-- End of BUFGMUX_1_inst instantiation
```

Verilog Instantiation Template

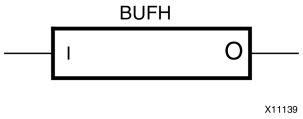
```
// BUFGMUX_1: Global Clock Mux Buffer with Output State 1
//            Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

BUFGMUX_1 #(
   .CLK_SEL_TYPE("SYNC")  // Glitchles ("SYNC") or fast ("ASYNC") clock switch-over
)
BUFGMUX_1_inst (
   .O(O),   // 1-bit output: Clock buffer output
   .I0(I0), // 1-bit input: Clock buffer input
   .I1(I1), // 1-bit input: Clock buffer input
   .S(S)    // 1-bit input: Clock buffer select
);

// End of BUFGMUX_1_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA Clocking Resources User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

BUFH

Primitive: Clock buffer for a single clocking region



Introduction

The BUFH primitive is provided to allow instantiation capability to access the HCLK clock buffer resources. The use of this component requires manual placement and special consideration and thus is recommended for more advanced users. Please refer to the *Spartan-6 FPGA Clocking Resources User Guide (UG382)* for details about using this component.

Port Descriptions

Port	Direction	Width	Function
I	Input	1	Clock Input
O	Output	1	Clock Output

Design Entry Method

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

VHDL Instantiation Template

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

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


-- BUFH: HROW Clock Buffer for a Single Clocking Region
--       Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of BUFH_inst instantiation
```

Verilog Instantiation Template

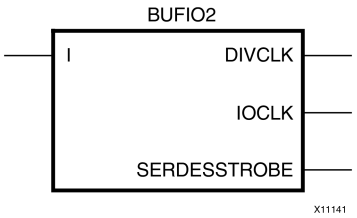
```
// BUFH: HROW Clock Buffer for a Single Clocking Region
//       Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

BUFH BUFH_inst (
   .O(O), // 1-bit output: Clock output
   .I(I)  // 1-bit input: Clock input
);

// End of BUFH_inst instantiation
```

For More Information

See the *Spartan-6 FPGA User Documentation (User Guides and Data Sheets)*.

BUFIO2

Primitive: Dual Clock Buffer and Strobe Pulse



Introduction

This primitive provides high-speed I/O clocking resources from an off-chip source intended to drive the synchronous I/O resources (ISERDES2, OSERDES2) and associated fabric resources via a BUFG with low skew. Please refer to the *Spartan-6 FPGA Clocking Resources User Guide (UG382)* for details about using this component.

Port Descriptions

Port	Direction	Width	Function
DIVCLK	Output	1	Divided clock output
I	Input	1	Clock input
IOCLK	Output	1	Clock output
SERDESSTROBE	Output	1	Output SERDES Strobe (connect to ISERDES/OSERDES)

Design Entry Method

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

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
DIVIDE	Decimal	1, 2, 3, 4, 5, 6, 7, 8	1	DIVCLK divider
DIVIDE_BYPASS	Boolean	TRUE, FALSE	TRUE	Bypass the divider circuitry
I_INVERT	Boolean	FALSE, TRUE	FALSE	Invert clock
USE_DOUBLER	Boolean	FALSE, TRUE	FALSE	Use doubler circuitry

VHDL Instantiation Template

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

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


-- BUFIO2: I/O Clock Buffer
--         Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

BUFIO2_inst : BUFIO2
generic map (
   DIVIDE => 1,           -- DIVCLK divider (1-8)
   DIVIDE_BYPASS => TRUE, -- Bypass the divider circuitry (TRUE/FALSE)
   I_INVERT => FALSE,     -- Invert clock (TRUE/FALSE)
   USE_DOUBLER => FALSE   -- Use doubler circuitry (TRUE/FALSE)
)
port map (
   DIVCLK => DIVCLK,             -- 1-bit output: Divided clock output
   IOCLK => IOCLK,               -- 1-bit output: I/O output clock
   SERDESSTROBE => SERDESSTROBE, -- 1-bit output: Output SERDES strobe (connect to ISERDES2/OSERDES2)
   I => I                        -- 1-bit input: Clock input (connect to IBUFG)
);

-- End of BUFIO2_inst instantiation
```

Verilog Instantiation Template

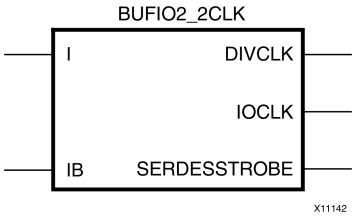
```
// BUFIO2: I/O Clock Buffer
//         Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

BUFIO2 #(
   .DIVIDE(1),             // DIVCLK divider (1-8)
   .DIVIDE_BYPASS("TRUE"), // Bypass the divider circuitry (TRUE/FALSE)
   .I_INVERT("FALSE"),     // Invert clock (TRUE/FALSE)
   .USE_DOUBLER("FALSE")   // Use doubler circuitry (TRUE/FALSE)
)
BUFIO2_inst (
   .DIVCLK(DIVCLK),             // 1-bit output: Divided clock output
   .IOCLK(IOCLK),               // 1-bit output: I/O output clock
   .SERDESSTROBE(SERDESSTROBE), // 1-bit output: Output SERDES strobe (connect to ISERDES2/OSERDES2)
   .I(I)                        // 1-bit input: Clock input (connect to IBUFG)
);

// End of BUFIO2_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA Clocking Resources User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

BUFIO2_2CLK

Primitive: Dual Clock Buffer and Strobe Pulse with Differential Input



Introduction

The BUFIO2_2CLK resource provides high-speed I/O clocking resources from an off-chip source intended to drive the synchronous I/O resources (ISERDES2, OSERDES2) and associated fabric resources via a BUFG with low skew. Please refer to the *Spartan-6 FPGA Clocking Resources User Guide (UG382)* for details about using this component.

Design Entry Method

Instantiation	Yes
Inference	No
CORE Generator™ and wizards	Yes
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;


-- BUFIO2_2CLK: Dual Input Differential Clock Buffer
--              Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

BUFIO2_2CLK_inst : BUFIO2_2CLK
generic map (
   DIVIDE => 2  -- DIVCLK divider (2-8)
)
port map (
   DIVCLK => DIVCLK,             -- 1-bit output: Divided clock output
   IOCLK => IOCLK,               -- 1-bit output: I/O output clock
   SERDESSTROBE => SERDESSTROBE, -- 1-bit output: Output SERDES strobe (connect to ISERDES2/OSERDES2)
   I => I,                       -- 1-bit input: Clock input (connect to IBUFG)
   IB => IB                      -- 1-bit input: Secondary clock input
);

-- End of BUFIO2_2CLK_inst instantiation
```

Verilog Instantiation Template

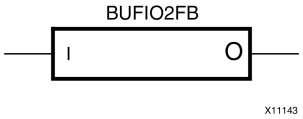
```
// BUFIO2_2CLK: Dual Input Differential Clock Buffer
//              Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

BUFIO2_2CLK #(
   .DIVIDE(2)  // DIVCLK divider (2-8)
)
BUFIO2_2CLK_inst (
   .DIVCLK(DIVCLK),             // 1-bit output: Divided clock output
   .IOCLK(IOCLK),               // 1-bit output: I/O output clock
   .SERDESSTROBE(SERDESSTROBE), // 1-bit output: Output SERDES strobe (connect to ISERDES2/OSERDES2)
   .I(I),                       // 1-bit input: Clock input (connect to IBUFG)
   .IB(IB)                      // 1-bit input: Secondary clock input
);

// End of BUFIO2_2CLK_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA Clocking Resources User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

BUFIO2FB

Primitive: Feedback Clock Buffer



Introduction

This element is a simple buffer that is delay matched to an associated BUFIO2 which is used for the feedback path for proper phase compensation of the feedback when using a DLL or PLL.

Port Descriptions

Port	Direction	Width	Function
I	Input	1	Input feedback clock.
O	Output	1	Output feedback clock (Connect to feedback input of DCM/PLL).

Design Entry Method

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

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
DIVIDE_BYPASS	Boolean	TRUE, FALSE	TRUE	Bypass Divider (TRUE/FALSE) Set the same as associated BUFIO2.

VHDL Instantiation Template

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

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


-- BUFIO2FB: DCM/PLL Feedback Clock Buffer
--           Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

BUFIO2FB_inst : BUFIO2FB
generic map (
   DIVIDE_BYPASS => TRUE  -- Bypass divider (TRUE/FALSE)
)
port map (
   O => O, -- 1-bit output: Output feedback clock (connect to feedback input of DCM/PLL)
   I => I  -- 1-bit input: Feedback clock input (connect to input port)
);

-- End of BUFIO2FB_inst instantiation
```

Verilog Instantiation Template

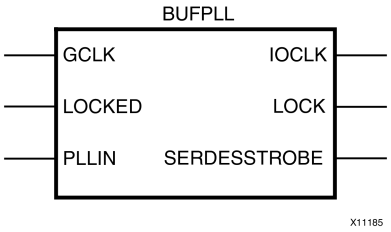
```
// BUFIO2FB: DCM/PLL Feedback Clock Buffer
//           Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

BUFIO2FB #(
   .DIVIDE_BYPASS("TRUE")  // Bypass divider (TRUE/FALSE)
)
BUFIO2FB_inst (
   .O(O), // 1-bit output: Output feedback clock (connect to feedback input of DCM/PLL)
   .I(I)  // 1-bit input: Feedback clock input (connect to input port)
);

// End of BUFIO2FB_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA Clocking Resources User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

BUFPLL

Primitive: PLL Buffer



Introduction

High-speed I/O clock buffer sourced from the PLL component.

Port Descriptions

Port	Direction	Width	Function
GCLK	Input	1	BUFG Clock Input.
IOCLK	Output	1	Output I/O Clock.
LOCK	Output	1	Synchronized LOCK output.
LOCKED	Input	1	LOCKED Input from PLL.
PLLIN	Input	1	Clock Input from PLL.
SERDESSTROBE	Output	1	SERDES strobe (connect to ISERDES/OSERDES).

Design Entry Method

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

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
DIVIDE	Integer	1, 2, 3, 4, 5, 6, 7, 8	1	DIVCLK Divider (1-8)
ENABLE_SYNC	Boolean	TRUE, FALSE	TRUE	Enable synchrnonization between PLL and GCLK (TRUE/FALSE).

VHDL Instantiation Template

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

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


-- BUFPLL: High-speed I/O PLL clock buffer
--         Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

BUFPLL_inst : BUFPLL
generic map (
   DIVIDE => 1,          -- DIVCLK divider (1-8)
   ENABLE_SYNC => TRUE  -- Enable synchrnonization between PLL and GCLK (TRUE/FALSE)
)
port map (
   IOCLK => IOCLK,               -- 1-bit output: Output I/O clock
   LOCK => LOCK,                 -- 1-bit output: Synchronized LOCK output
   SERDESSTROBE => SERDESSTROBE, -- 1-bit output: Output SERDES strobe (connect to ISERDES2/OSERDES2)
   GCLK => GCLK,                 -- 1-bit input: BUFG clock input
   LOCKED => LOCKED,             -- 1-bit input: LOCKED input from PLL
   PLLIN => PLLIN                -- 1-bit input: Clock input from PLL
);

-- End of BUFPLL_inst instantiation
```

Verilog Instantiation Template

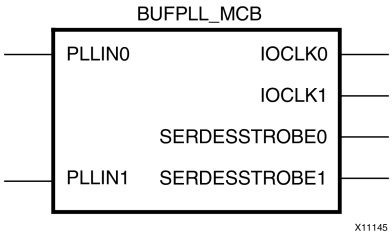
```
// BUFPLL: High-speed I/O PLL clock buffer
//         Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

BUFPLL #(
   .DIVIDE(1),           // DIVCLK divider (1-8)
   .ENABLE_SYNC("TRUE")  // Enable synchrnonization between PLL and GCLK (TRUE/FALSE)
)
BUFPLL_inst (
   .IOCLK(IOCLK),               // 1-bit output: Output I/O clock
   .LOCK(LOCK),                 // 1-bit output: Synchronized LOCK output
   .SERDESSTROBE(SERDESSTROBE), // 1-bit output: Output SERDES strobe (connect to ISERDES2/OSERDES2)
   .GCLK(GCLK),                 // 1-bit input: BUFG clock input
   .LOCKED(LOCKED),             // 1-bit input: LOCKED input from PLL
   .PLLIN(PLLIN)                // 1-bit input: Clock input from PLL
);

// End of BUFPLL_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA Clocking Resources User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

BUFPLL_MCB

Primitive: PLL Buffer for the Memory Controller Block



Introduction

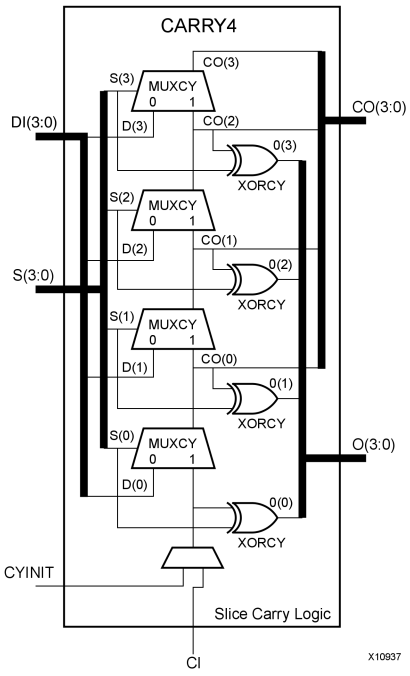
The BUFPLL_MCB is a component used by the Memory Interface Generator (MIG) core in conjunction with the MCB block to implement external memory interfaces. The use of this block outside of MIG is not supported.

For More Information

- See the *Xilinx Memory Interface Solutions User Guide*.
- See the *Spartan-6 FPGA Clocking Resources User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

CARRY4

Primitive: Fast Carry Logic with Look Ahead



Introduction

This circuit design represents the fast carry logic for a slice. The carry chain consists of a series of four MUXes and four XORs that connect to the other logic (LUTs) in the slice via dedicated routes to form more complex functions. The fast carry logic is useful for building arithmetic functions like adders, counters, subtractors and add/subs, as well as such other logic functions as wide comparators, address decoders, and some logic gates (specifically, AND and OR).

Port Descriptions

Port	Direction	Width	Function
O	Output	4	Carry chain XOR general data out
CO	Output	4	Carry-out of each stage of the carry chain
DI	Input	4	Carry-MUX data input
S	Input	4	Carry-MUX select line
CYINIT	Input	1	Carry-in initialization input
CI	Input	1	Carry cascade input

Design Entry Method

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

VHDL Instantiation Template

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

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


-- CARRY4: Fast Carry Logic Component
--         Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

CARRY4_inst : CARRY4
port map (
   CO => CO,         -- 4-bit carry out
   O => O,           -- 4-bit carry chain XOR data out
   CI => CI,         -- 1-bit carry cascade input
   CYINIT => CYINIT, -- 1-bit carry initialization
   DI => DI,         -- 4-bit carry-MUX data in
   S => S            -- 4-bit carry-MUX select input
);

-- End of CARRY4_inst instantiation
```

Verilog Instantiation Template

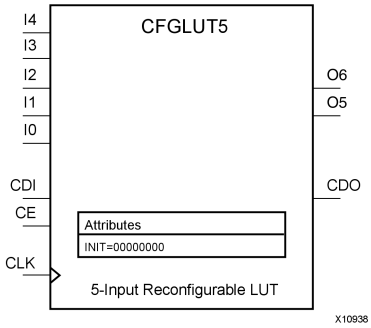
```
// CARRY4: Fast Carry Logic Component
//         Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

CARRY4 CARRY4_inst (
   .CO(CO),         // 4-bit carry out
   .O(O),           // 4-bit carry chain XOR data out
   .CI(CI),         // 1-bit carry cascade input
   .CYINIT(CYINIT), // 1-bit carry initialization
   .DI(DI),         // 4-bit carry-MUX data in
   .S(S)            // 4-bit carry-MUX select input
);

// End of CARRY4_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA Configurable Logic Block User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

CFGLUT5

Primitive: 5-input Dynamically Reconfigurable Look-Up Table (LUT)



Introduction

This element is a runtime, dynamically reconfigurable, 5-input look-up table (LUT) that enables the changing of the logical function of the LUT during circuit operation. Using the CDI pin, a new INIT value can be synchronously shifted in serially to change the logical function. The O6 output pin produces the logical output function, based on the current INIT value loaded into the LUT and the currently selected I0-I4 input pins. Optionally, you can use the O5 output in combination with the O6 output to create two individual 4-input functions sharing the same inputs or a 5-input function and a 4-input function that uses a subset of the 5-input logic (see tables below). This component occupies one of the four LUT6 components within a Slice-M.

To cascade this element, connect the CDO pin from each element to the CDI input of the next element. This will allow a single serial chain of data (32-bits per LUT) to reconfigure multiple LUTs.

Port Descriptions

Port	Direction	Width	Function
O6	Output	1	5-LUT output
O5	Output	1	4-LUT output
I0, I1, I2, I3, I4	Input	1	LUT inputs
CDO	Output	1	Reconfiguration data cascaded output (optionally connect to the CDI input of a subsequent LUT)
CDI	Input	1	Reconfiguration data serial input
CLK	Input	1	Reconfiguration clock
CE	Input	1	Active high reconfiguration clock enable

Design Entry Method

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

- Connect the CLK input to the clock source used to supply the reconfiguration data.
- Connect the CDI input to the source of the reconfiguration data.
- Connect the CE pin to the active high logic if you need to enable/disable LUT reconfiguration.
- Connect the I4-I0 pins to the source inputs to the logic equation. The logic function is output on O6 and O5.
- To cascade this element, connect the CDO pin from each element to the CDI input of the next element to allow a single serial chain of data to reconfigure multiple LUTs.

The INIT attribute should be placed on this design element to specify the initial logical function of the LUT. A new INIT can be loaded into the LUT any time during circuit operation by shifting in 32-bits per LUT in the chain, representing the new INIT value. Disregard the O6 and O5 output data until all 32-bits of new INIT data has been clocked into the LUT. The logical function of the LUT changes as new INIT data is shifted into it. Data should be shifted in MSB (INIT[31]) first and LSB (INIT[0]) last.

In order to understand the O6 and O5 logical value based on the current INIT, see the table below:

I4 I3 I2 I1 I0	O6 Value	O5 Value
1 1 1 1 1	INIT[31]	INIT[15]
1 1 1 1 0	INIT[30]	INIT[14]
. . .	. . .	. . .
1 0 0 0 1	INIT[17]	INIT[1]
1 0 0 0 0	INIT[16]	INIT[0]
0 1 1 1 1	INIT[15]	INIT[15]
0 1 1 1 0	INIT[14]	INIT[14]
. . .	. . .	. . .
0 0 0 0 1	INIT[1]	INIT[1]
0 0 0 0 0	INIT[0]	INIT[0]

For instance, the INIT value of FFFF8000 would represent the following logical equations:

- O6 = I4 or (I3 and I2 and I1 and I0)
- O5 = I3 and I2 and I1 and I0

To use these elements as two, 4-input LUTs with the same inputs but different functions, tie the I4 signal to a logical one. The INIT[31:16] values apply to the logical values of the O6 output and INIT [15:0] apply to the logical values of the O5 output.

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
INIT	Hexadecimal	Any 32-bit Value	All zeros	Specifies the initial logical expression of this 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;


-- CFGLUT5: Reconfigurable 5-input LUT
--          Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of CFGLUT5_inst instantiation
```

Verilog Instantiation Template

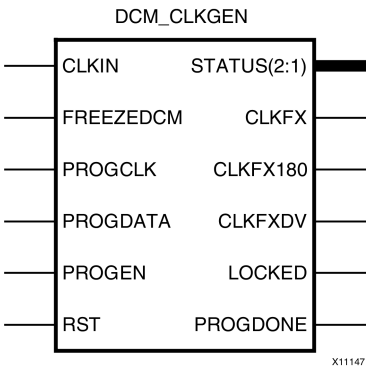
```
// CFGLUT5: Reconfigurable 5-input LUT
//          Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

CFGLUT5 #(
   .INIT(32'h00000000) // Specify initial LUT contents
) CFGLUT5_inst (
   .CDO(CDO), // Reconfiguration cascade output
   .O5(O5),   // 4-LUT output
   .O6(O6),   // 5-LUT output
   .CDI(CDI), // Reconfiguration data input
   .CE(CE),   // Reconfiguration enable input
   .CLK(CLK), // Clock input
   .I0(I0),   // Logic data input
   .I1(I1),   // Logic data input
   .I2(I2),   // Logic data input
   .I3(I3),   // Logic data input
   .I4(I4)    // Logic data input
);

// End of CFGLUT5_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA Configurable Logic Block User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

DCM_CLKGEN

Primitive: Digital Clock Manager.



Introduction

Digital Clock Manager (DCM) is set to frequency aligned mode and thus is not phase aligned (no phase relationship) to the input clock. By being in frequency aligned mode, it allows additional capabilities including programmable output clock synthesis, jitter reduction, spread spectrum, and free running oscillator modes. Please refer to the *Spartan-6 FPGA Clocking Resources User Guide (UG382)* for details in using this component.

Port Descriptions

Port	Direction	Width	Function
CLKFX	Output	1	Synthesized clock output, controlled by the CLKFX_MULTIPLY and CLKFX_DIVIDE attributes. Can be either statically set or dynamically programmed through a dedicated 4-wire SPI port (PROGDATA, PROGCLK, PROGDONE, and PROGEN). Always has a 50% duty cycle.
CLKFXDV	Output	1	Divided CLKFX output clock. Divide value derived from CLKFXDV_DIVIDE attribute. There is no phase alignment between CLKFX and CLKFXDV.
CLKFX180	Output	1	Synthesized clock output CLKFX, 180 phase shift (appears to be an inverted version of CLKFX). Always has a 50% duty cycle.
CLKIN	Input	1	Clock input to DCM. Always required. The CLKIN frequency and jitter must fall within the limits specified in the data sheet. In the case of free-running oscillator mode, running clock needs to be connected until DCM is locked and DCM is frozen, then clock can be removed. In the other modes, a free running clock needs to be provided and remains.
FREEZEDCM	Input	1	Prevents tap adjustment drift in the event of a lost CLKIN input. The DCM is then configured into a free-run mode.
LOCKED	Output	1	Synchronous output indicates whether the DCM is ready for operation. • 0 - DCM clock outputs are not valid • 1 - DCM is ready for operation • 1-to-0 - DCM lost LOCK. Reset DCM
PROGCLK	Input	1	Clock input for M and/or D reconfiguration.
PROGDATA	Input	1	Serial data input to supply information for the reprogramming of M and D values of the DCM. This input must be applied synchronous to the PROGCLK input.

Port	Direction	Width	Function
PROGDONE	Output	1	Active high output to indicate the successful reprogramming of an M or D value.
PROGEN	Input	1	Active high enable input for the reprogramming of M/D values. This input must be applied synchronously to the PROGCLK input.
RST	Input	1	Resets the DCM circuitry. The RST signal is an active High asynchronous reset. Asserting the RST signal asynchronously forces all DCM outputs Low (the LOCKED signal, all status signals, and all output clocks within four source clock cycles). Because the reset is asynchronous, the last cycle of the clocks can exhibit an unintended short pulse, severely distorted duty-cycle, and no longer phase adjust with respect to one another while deasserting. The RST pin must be used when reconfiguring the device or changing the input frequency. Deasserting the RST signal synchronously starts the locking process at the next CLKIN cycle. To ensure a proper DCM reset and locking process, the RST signal must be deasserted after the CLKIN signal has been present and stable for at least three clock cycles. In all designs, the DCM must be held in reset until the clock is stable. During configuration, the DCM is automatically held in reset until GSR is released. If the clock is stable when GSR is released, DCM reset after configuration is not necessary.
STATUS[2:1]	Output	2	Clock Status lines. • STATUS[1] indicates that CLKIN has stopped. • STATUS[2] indicates that CLKFX or CLKFX180 has stopped.

Design Entry Method

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

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
CLKFX_DIVIDE	Integer	1 to 256	1	This value in conjunction with the input frequency and CLKFX_MULTIPLY value, determines the resultant output frequency for the CLKFX and CLKFX180 outputs.
CLKFXDV_DIVIDE	Integer	2, 4, 8, 16, 32	2	Specifies divide value for CLKFXDV.
CLKFX_MD_MAX	3 significant digit Float	0.000 to 256.000	0.000	When using the DCM_CLKGEN with variable M and D values, this would specify the maximum ratio of M and D used during static timing analysis.
CLKFX_MULTIPLY	Integer	2 to 256	4	This value in conjunction with the input frequency and CLKFX_DIVIDE value, determine

Attribute	Data Type	Allowed Values	Default	Description
				the resultant output frequency for the CLKFX and CLKFX180 outputs.
CLKIN_PERIOD	Float	2.000 to 1000.00	None	This attribute specifies the source clock period which is used to help the DCM adjust for the optimum CLKFX/CLKFX180 outputs and also result in faster locking time.
DFS_BANDWIDTH	String	"OPTIMIZED", "HIGH", "LOW"	"OPTIMIZED"	Specifies the frequency adjust bandwidth of the DCM due to process, voltage, and temperature (PVT).
PROG_MD_ BANDWIDTH	String	"OPTIMIZED", "HIGH", "LOW"	"OPTIMIZED"	Specifies the frequency adjust bandwidth of the DCM due to change of programming of the M and D values.
SPREAD_SPECTRUM	String	"NONE", "CENTER_LOW_ SPREAD", "CENTER_HIGH_ SPREAD", "VIDEO_LINK_M0", "VIDEO_LINK_M1", "VIDEO_LINK_M2"	"NONE"	Specifies a supported mode for Spread Spectrum. Must be used in conjunction with the appropriate IP to fully realize the frequency hopping. Used for fixed spread spectrum ("CENTER_LOW_SPREAD" and "CENTER_HIGH_SPREAD") or soft spread spectrum ("VIDEO_LINK_M0", "VIDEO_LINK_M1", and "VIDEO_LINK_M2"). Soft spread spectrum must be used in conjunction with the soft spread spectrum reference design.
STARTUP_WAIT	Boolean	FALSE, TRUE	FALSE	Delays the configuration DONE signal until DCM LOCKED signal goes high.

VHDL 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_CLKGEN: Frequency Aligned Digital Clock Manager
--             Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

DCM_CLKGEN_inst : DCM_CLKGEN
generic map (
   CLKFXDV_DIVIDE => 2,       -- CLKFXDV divide value (2, 4, 8, 16, 32)
   CLKFX_DIVIDE => 1,         -- Divide value - D - (1-256)
   CLKFX_MD_MAX => 0.0,       -- Specify maximum M/D ratio for timing anlysis
   CLKFX_MULTIPLY => 4,       -- Multiply value - M - (2-256)
   CLKIN_PERIOD => 0.0,       -- Input clock period specified in nS
   SPREAD_SPECTRUM => "NONE", -- Spread Spectrum mode "NONE", "CENTER_LOW_SPREAD", "CENTER_HIGH_SPREAD",
                              -- "VIDEO_LINK_M0", "VIDEO_LINK_M1" or "VIDEO_LINK_M2"
   STARTUP_WAIT => FALSE      -- Delay config DONE until DCM_CLKGEN LOCKED (TRUE/FALSE)
)
port map (
   CLKFX => CLKFX,         -- 1-bit output: Generated clock output
   CLKFX180 => CLKFX180,   -- 1-bit output: Generated clock output 180 degree out of phase from CLKFX.
   CLKFXDV => CLKFXDV,     -- 1-bit output: Divided clock output
```

```
   LOCKED => LOCKED,       -- 1-bit output: Locked output
   PROGDONE => PROGDONE,   -- 1-bit output: Active high output to indicate the successful re-programming
   STATUS => STATUS,       -- 2-bit output: DCM_CLKGEN status
   CLKIN => CLKIN,         -- 1-bit input: Input clock
   FREEZEDCM => FREEZEDCM, -- 1-bit input: Prevents frequency adjustments to input clock
   PROGCLK => PROGCLK,     -- 1-bit input: Clock input for M/D reconfiguration
   PROGDATA => PROGDATA,   -- 1-bit input: Serial data input for M/D reconfiguration
   PROGEN => PROGEN,       -- 1-bit input: Active high program enable
   RST => RST              -- 1-bit input: Reset input pin
);

-- End of DCM_CLKGEN_inst instantiation
```

Verilog Instantiation Template

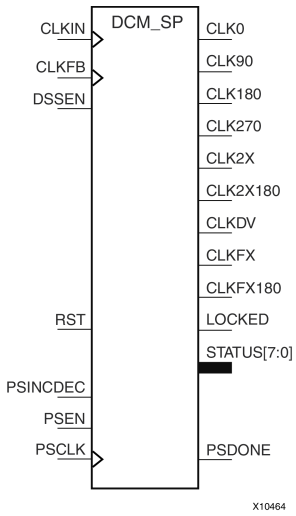
```
// DCM_CLKGEN: Frequency Aligned Digital Clock Manager
//             Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

DCM_CLKGEN #(
   .CLKFXDV_DIVIDE(2),       // CLKFXDV divide value (2, 4, 8, 16, 32)
   .CLKFX_DIVIDE(1),         // Divide value - D - (1-256)
   .CLKFX_MD_MAX(0.0),       // Specify maximum M/D ratio for timing anlysis
   .CLKFX_MULTIPLY(4),       // Multiply value - M - (2-256)
   .CLKIN_PERIOD(0.0),       // Input clock period specified in nS
   .SPREAD_SPECTRUM("NONE"), // Spread Spectrum mode "NONE", "CENTER_LOW_SPREAD", "CENTER_HIGH_SPREAD",
                             // "VIDEO_LINK_M0", "VIDEO_LINK_M1" or "VIDEO_LINK_M2"
   .STARTUP_WAIT("FALSE")    // Delay config DONE until DCM_CLKGEN LOCKED (TRUE/FALSE)
)
DCM_CLKGEN_inst (
   .CLKFX(CLKFX),          // 1-bit output: Generated clock output
   .CLKFX180(CLKFX180),    // 1-bit output: Generated clock output 180 degree out of phase from CLKFX.
   .CLKFXDV(CLKFXDV),      // 1-bit output: Divided clock output
   .LOCKED(LOCKED),        // 1-bit output: Locked output
   .PROGDONE(PROGDONE),    // 1-bit output: Active high output to indicate the successful re-programming
   .STATUS(STATUS),        // 2-bit output: DCM_CLKGEN status
   .CLKIN(CLKIN),          // 1-bit input: Input clock
   .FREEZEDCM(FREEZEDCM), // 1-bit input: Prevents frequency adjustments to input clock
   .PROGCLK(PROGCLK),      // 1-bit input: Clock input for M/D reconfiguration
   .PROGDATA(PROGDATA),    // 1-bit input: Serial data input for M/D reconfiguration
   .PROGEN(PROGEN),        // 1-bit input: Active high program enable
   .RST(RST)               // 1-bit input: Reset input pin
);

// End of DCM_CLKGEN_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA Clocking Resources User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

DCM_SP

Primitive: Digital Clock Manager



Introduction

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

Port Descriptions

Port	Direction	Width	Function
CLKDV	Output	1	Divided clock output, controlled by the CLKDV_DIVIDE attribute. The CLKDV output has a 50% duty cycle unless the CLKDV_DIVIDE attribute is a non-integer value.
CLKFB	Input	1	Clock feedback input to DCM. The feedback input is required unless the DFS outputs, CLKFX or CLKFX180, are used standalone. The source of the CLKFB input must be the CLK0 or CLK2X output from the DCM and the CLK_FEEDBACK must be set to 1X or 2X accordingly. When set to NONE, CLKFB is unused and should be tied low. Ideally, the feedback point includes the delay added by the clock distribution network, either internally or externally.
CLKFX	Output	1	Synthesized clock output, controlled by the CLKFX_MULTIPLY and CLKFX_DIVIDE attributes. Always has a 50% duty cycle. If no phase relationship is necessary, then no clock feedback is required.
CLKFX180	Output	1	Synthesized clock output CLKFX, 180 degree phase shift (an inverted version of CLKFX). Always has a 50% duty cycle. If no phase relationship is necessary, then no feedback loop is required.
CLKIN	Input	1	Clock input to DCM. Always required. The CLKIN frequency and jitter must fall within the limits specified in the data sheet.
CLK0	Output	1	Same frequency as CLKIN, 0 phase shift (i.e., not phase shifted). Always conditioned to a 50% duty cycle on Spartan®-6 FPGAs. CLK_FEEDBACK must be set to 1X or 2X to deskew CLK0.
CLK2X	Output	1	Double-frequency clock output, 0 degree phase shift. When available, the CLK2X output always has a 50% duty cycle. Either CLK0 or CLK2X is required as a feedback source for DLL functions.

Port	Direction	Width	Function
CLK2X180	Output	1	Double-frequency clock output, 180 degree phase shift. When available, the CLK2X180 output always has a 50% duty cycle.
CLK90	Output	1	Same frequency as CLKIN, 90 degree phase shift (quarter period). Always conditioned to a 50% duty cycle on Spartan®-6 FPGAs.
CLK180	Output	1	Same frequency as CLKIN, 180 degree phase shift (half period). Always conditioned to a 50% duty cycle on Spartan®-6 FPGAs.
CLK270	Output	1	Same frequency as CLKIN, 270 degree phase shift (three-quarters period). Always conditioned to a 50% duty cycle on Spartan®-6 FPGAs.
LOCKED	Output	1	All DCM features have locked onto the CLKIN frequency. Clock outputs are now valid, assuming CLKIN is within specified limits. • 0 - DCM is attempting to lock onto CLKIN frequency. DCM clock outputs are not valid. • 1 - DCM is locked onto CLKIN frequency. DCM clock outputs are valid. • 1-to-0 - DCM lost lock. Reset DCM.
PSCLK	Input	1	Clock input to variable phase shifter, clocked on rising edge. When using a global clock buffer, only the upper eight BUFGMUXs can drive PSCLK: BUFGMUX_X2Y1, BUFGMUX_X2Y2, BUFGMUX_X2Y3, BUFGMUX_X2Y4, BUFGMUX_X3Y5, BUFGMUX_X3Y6, BUFGMUX_X3Y7 and BUFGMUX_X3Y8.
PSDONE	Output	1	Variable phase shift operation complete. • 0 - No phase shift operation is active or phase shift operation is in progress. • 1 - Requested phase shift operation is complete. Output High for one PSCLK cycle. Next variable phase shift operation can commence.
PSEN	Input	1	Variable phase-shift enable. Can be inverted within a DCM block. Non-inverted behavior shown below. • 0 - Disable variable phase shift. Ignore inputs to phase shifter. • 1 - Enable variable phase shift operations on next rising PSCLK clock edge. **Note** Tie to 0 when not in use.
PSINCDEC	Input	1	Increment/decrement variable phase shift. Can be inverted within a DCM block. Non-inverted behavior shown below. • 0 - Decrement phase shift value on next enabled, rising PSCLK clock edge. • 1 - Increment phase shift value on next enabled, rising PSCLK clock edge.

Port	Direction	Width	Function
RST	Input	1	Asynchronous reset input. Resets the DCM logic to its postconfiguration state. Causes DCM to reacquire and relock to the CLKIN input. Invertible within DCM block. Non-inverted behavior shown below. • 0 - No effect. • 1 - Reset DCM block. Hold RST pulse High for at least three valid CLKIN cycles.
STATUS[7:0]	Output	8	The status output bus provides DCM status. • STATUS[0] - Variable phase shift overflow. Control output for variable fine phase shifting. The variable phase shifter has reached a minimum or maximum limit value. The limit value is either +/-255 or a lesser value if the phase shift has reached the end of the delay line. – 0 - The phase shift has not yet reached its limit value. – 1 - The phase shift has reached its limited value. • STATUS[1] - CLKIN Input Stopped Indicator. Available only when the CLKFB feedback input is connected. Held in reset until the LOCKED output is asserted. Requires at least one CLKIN cycle to become active. Never asserted if CLKIN never toggles. – 0 - CLKIN input is toggling. – 1 - CLKIN input is not toggling even though the locked output can still be High. • STATUS[2] - CLKFX or CLKFX180 output stopped indicator. – 0 - CLKFX and CLKFX180 outputs are toggling. – 1 - CLKFX and CLKFX180 outputs are not toggling, even though the LOCKED output can still be High. • STATUS[4:3] - Reserved. • STATUS[5] - A mirrored version of CLKFX during DCM_SP lock period status. • STATUS[6] - Reserved. • STATUS[7] - A mirrored version of CLKIN during DCM_SP lock period status.

Design Entry Method

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

Available Attributes

Attribute	Data Type	Allowed_Values	Default	Description
CLK_FEEDBACK	String	"1X", "2X", "NONE"	"1X"	Defines the DCM feedback mode. • "1X" - CLK0 as feedback. • "2X" - CLK2X as feedback.
CLKDV_DIVIDE	1 significant digit Float	2.0, 1.5, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0, 5.5, 6.0, 6.5, 7.0, 7.5, 8.0, 9.0, 10.0, 11.0, 12.0, 13.0, 14.0, 15.0, 16.0	2.0	Specifies the extent to which the CLKDLL, CLKDLLE, CLKDLLHF, or DCM_SP clock divider (CLKDV output) is to be frequency divided.
CLKFX_DIVIDE	Integer	1 to 32	1	Specifies the frequency divider value for the CLKFX output.
CLKFX_MULTIPLY	Integer	2 to 32	4	Specifies the frequency multiplier value for the CLKFX output.
CLKIN_DIVIDE_BY_2	Boolean	FALSE, TRUE	FALSE	Enables CLKIN divide by two features.
CLKIN_PERIOD	Float	2.000 to 1000.000	None	Specifies the input period to the DCM_SP CLKIN input in ns.
CLKOUT_PHASE_SHIFT	String	"NONE", "FIXED", "VARIABLE"	"NONE"	This attribute specifies the phase shift mode. • "NONE" - No phase shift capability. Any set value has no effect. • "FIXED" - DCM outputs are a fixed phase shift from CLKIN. Value is specified by PHASE_SHIFT attribute. • "VARIABLE" - Allows the DCM outputs to be shifted in a positive and negative range relative to CLKIN. Starting value is specified by PHASE_SHIFT.
DESKEW_ADJUST	String	"SYSTEM_SYNCHRONOUS", "SOURCE_SYNCHRONOUS"	"SYSTEM_SYNCHRONOUS"	Sets configuration bits affecting the clock delay alignment between the DCM_SP output clocks and an FPGA clock input pin.
DFS_FREQUENCY_MODE	String	"LOW", "HIGH"	"LOW"	This is a legacy attribute. The DCM is always in the automatic frequency search mode. Setting High or Low makes no effect.
DLL_FREQUENCY_MODE	String	"LOW", "HIGH"	"LOW"	This is a legacy attribute. The DCM is always in the automatic frequency search mode. Setting High or Low makes no effect.
DUTY_CYCLE_CORRECTION	Boolean	TRUE, FALSE	TRUE	Unsupported
FACTORY_JF	Hexadecimal	16'h8080 to 16'hffff	16'hc080	Unsupported

Attribute	Data Type	Allowed_Values	Default	Description
PHASE_SHIFT	Integer	-255 to 255	0	The PHASE_SHIFT attribute is applicable only if the CLKOUT_PHASE_SHIFT attribute is set to FIXED or VARIABLE. Defines the rising-edge skew between CLKIN and all the DCM clock outputs at configuration and consequently phase shifts the DCM clock outputs. The skew or phase shift value is specified as an integer that represents a fraction of the clock period as expressed in the equations in Fine Phase Shifting. Actual allowable values depends on input clock frequency. The actual range is less when TCLKIN > FINE_SHIFT_RANGE. The FINE_SHIFT_RANGE specification represents the total delay of all taps in the delay line.
STARTUP_WAIT	Boolean	FALSE, TRUE	FALSE	Controls whether the FPGA configuration signal DONE waits for the DCM to assert its LOCKED signal before going High. • FALSE - Default. DONE is asserted at the end of configuration without waiting for the DCM to assert LOCKED. • TRUE - The DONE signal does not transition High until the LOCKED signal transitions High on the associated DCM. STARTUP_WAIT does not prevent LOCKED from transitioning High. The FPGA startup sequence must also be modified to insert a LCK (lock) cycle before the postponed cycle. The DONE cycle or the GWE cycle are typical choices. When more than one DCM is configured, the FPGA waits until all DCMs are LOCKED.

VHDL 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
--         Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

DCM_SP_inst : DCM_SP
generic map (
   CLKDV_DIVIDE => 2.0,                   -- CLKDV divide value
                                       -- (1.5,2,2.5,3,3.5,4,4.5,5,5.5,6,6.5,7,7.5,8,9,10,11,12,13,14,15,16).
   CLKFX_DIVIDE => 1,                     -- Divide value on CLKFX outputs - D - (1-32)
   CLKFX_MULTIPLY => 4,                   -- Multiply value on CLKFX outputs - M - (2-32)
   CLKIN_DIVIDE_BY_2 => FALSE,            -- CLKIN divide by two (TRUE/FALSE)
   CLKIN_PERIOD => 10.0,                  -- Input clock period specified in nS
   CLKOUT_PHASE_SHIFT => "NONE",          -- Output phase shift (NONE, FIXED, VARIABLE)
   CLK_FEEDBACK => "1X",                  -- Feedback source (NONE, 1X, 2X)
   DESKEW_ADJUST => "SYSTEM_SYNCHRONOUS", -- SYSTEM_SYNCHRNOUS or SOURCE_SYNCHRONOUS
   DFS_FREQUENCY_MODE => "LOW",           -- Unsupported - Do not change value
```

```
   DLL_FREQUENCY_MODE => "LOW",           -- Unsupported - Do not change value
   DSS_MODE => "NONE",                    -- Unsupported - Do not change value
   DUTY_CYCLE_CORRECTION => TRUE,         -- Unsupported - Do not change value
   FACTORY_JF => X"c080",                 -- Unsupported - Do not change value
   PHASE_SHIFT => 0,                      -- Amount of fixed phase shift (-255 to 255)
   STARTUP_WAIT => FALSE                  -- Delay config DONE until DCM_SP LOCKED (TRUE/FALSE)
)
port map (
   CLK0 => CLK0,         -- 1-bit output: 0 degree clock output
   CLK180 => CLK180,     -- 1-bit output: 180 degree clock output
   CLK270 => CLK270,     -- 1-bit output: 270 degree clock output
   CLK2X => CLK2X,       -- 1-bit output: 2X clock frequency clock output
   CLK2X180 => CLK2X180, -- 1-bit output: 2X clock frequency, 180 degree clock output
   CLK90 => CLK90,       -- 1-bit output: 90 degree clock output
   CLKDV => CLKDV,       -- 1-bit output: Divided clock output
   CLKFX => CLKFX,       -- 1-bit output: Digital Frequency Synthesizer output (DFS)
   CLKFX180 => CLKFX180, -- 1-bit output: 180 degree CLKFX output
   LOCKED => LOCKED,     -- 1-bit output: DCM_SP Lock Output
   PSDONE => PSDONE,     -- 1-bit output: Phase shift done output
   STATUS => STATUS,     -- 8-bit output: DCM_SP status output
   CLKFB => CLKFB,       -- 1-bit input: Clock feedback input
   CLKIN => CLKIN,       -- 1-bit input: Clock input
   DSSEN => DSSEN,       -- 1-bit input: Unsupported, specify to GND.
   PSCLK => PSCLK,       -- 1-bit input: Phase shift clock input
   PSEN => PSEN,         -- 1-bit input: Phase shift enable
   PSINCDEC => PSINCDEC, -- 1-bit input: Phase shift increment/decrement input
   RST => RST            -- 1-bit input: Active high reset input
);

-- End of DCM_SP_inst instantiation
```

Verilog Instantiation Template

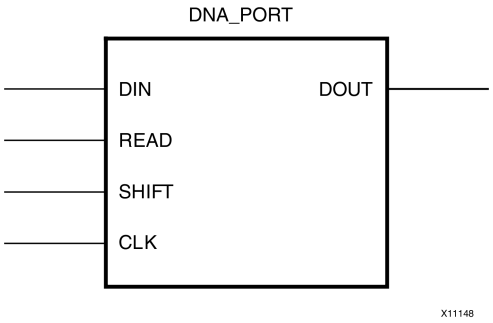
```
// DCM_SP: Digital Clock Manager
//         Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

DCM_SP #(
   .CLKDV_DIVIDE(2.0),                   // CLKDV divide value
                                 // (1.5,2,2.5,3,3.5,4,4.5,5,5.5,6,6.5,7,7.5,8,9,10,11,12,13,14,15,16).
   .CLKFX_DIVIDE(1),                     // Divide value on CLKFX outputs - D - (1-32)
   .CLKFX_MULTIPLY(4),                   // Multiply value on CLKFX outputs - M - (2-32)
   .CLKIN_DIVIDE_BY_2("FALSE"),          // CLKIN divide by two (TRUE/FALSE)
   .CLKIN_PERIOD(10.0),                  // Input clock period specified in nS
   .CLKOUT_PHASE_SHIFT("NONE"),          // Output phase shift (NONE, FIXED, VARIABLE)
   .CLK_FEEDBACK("1X"),                  // Feedback source (NONE, 1X, 2X)
   .DESKEW_ADJUST("SYSTEM_SYNCHRONOUS"), // SYSTEM_SYNCHRNOUS or SOURCE_SYNCHRONOUS
   .DFS_FREQUENCY_MODE("LOW"),           // Unsupported - Do not change value
   .DLL_FREQUENCY_MODE("LOW"),           // Unsupported - Do not change value
   .DSS_MODE("NONE"),                    // Unsupported - Do not change value
   .DUTY_CYCLE_CORRECTION("TRUE"),       // Unsupported - Do not change value
   .FACTORY_JF(16'hc080),                // Unsupported - Do not change value
   .PHASE_SHIFT(0),                      // Amount of fixed phase shift (-255 to 255)
   .STARTUP_WAIT("FALSE")                // Delay config DONE until DCM_SP LOCKED (TRUE/FALSE)
)
DCM_SP_inst (
   .CLK0(CLK0),         // 1-bit output: 0 degree clock output
   .CLK180(CLK180),     // 1-bit output: 180 degree clock output
   .CLK270(CLK270),     // 1-bit output: 270 degree clock output
   .CLK2X(CLK2X),       // 1-bit output: 2X clock frequency clock output
   .CLK2X180(CLK2X180), // 1-bit output: 2X clock frequency, 180 degree clock output
   .CLK90(CLK90),       // 1-bit output: 90 degree clock output
   .CLKDV(CLKDV),       // 1-bit output: Divided clock output
   .CLKFX(CLKFX),       // 1-bit output: Digital Frequency Synthesizer output (DFS)
   .CLKFX180(CLKFX180), // 1-bit output: 180 degree CLKFX output
   .LOCKED(LOCKED),     // 1-bit output: DCM_SP Lock Output
   .PSDONE(PSDONE),     // 1-bit output: Phase shift done output
   .STATUS(STATUS),     // 8-bit output: DCM_SP status output
   .CLKFB(CLKFB),       // 1-bit input: Clock feedback input
   .CLKIN(CLKIN),       // 1-bit input: Clock input
   .DSSEN(DSSEN),       // 1-bit input: Unsupported, specify to GND.
   .PSCLK(PSCLK),       // 1-bit input: Phase shift clock input
   .PSEN(PSEN),         // 1-bit input: Phase shift enable
   .PSINCDEC(PSINCDEC), // 1-bit input: Phase shift increment/decrement input
   .RST(RST)            // 1-bit input: Active high reset input
);

// End of DCM_SP_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA Clocking Resources User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

DNA_PORT

Primitive: Device DNA Data Access Port



Introduction

This element allows access to a dedicated shift register that can be loaded with the Device DNA data bits (unique ID) for a given device. In addition to shifting out the DNA data bits, this component allows for the inclusion of supplemental bits of your 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. 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 shifted 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.

Port Descriptions

Port	Direction	Width	Function
CLK	Input	1	Clock input.
DIN	Input	1	User data input pin.
DOUT	Output	1	DNA output data.
READ	Input	1	Active high load DNA, active low read input.
SHIFT	Input	1	Active high shift enable input.

Design Entry Method

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

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

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
SIM_DNA_VALUE	Hexa-decimal	57'h00000000 0000000 to 57'h1fffffffffffff	57'h00000000 0000000	Specifies the Pre-programmed factory ID 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;


-- DNA_PORT: Device DNA Data Access Port
--           Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

DNA_PORT_inst : DNA_PORT
generic map (
   SIM_DNA_VALUE => X"000000000000000"  -- Specifies the Pre-programmed factory ID value
)
port map (
   DOUT => DOUT,   -- 1-bit output: DNA output data
   CLK => CLK,     -- 1-bit input: Clock input
   DIN => DIN,     -- 1-bit input: User data input pin
   READ => READ,   -- 1-bit input: Active high load DNA, active low read input
   SHIFT => SHIFT  -- 1-bit input: Active high shift enable input
);

-- End of DNA_PORT_inst instantiation
```

Verilog Instantiation Template

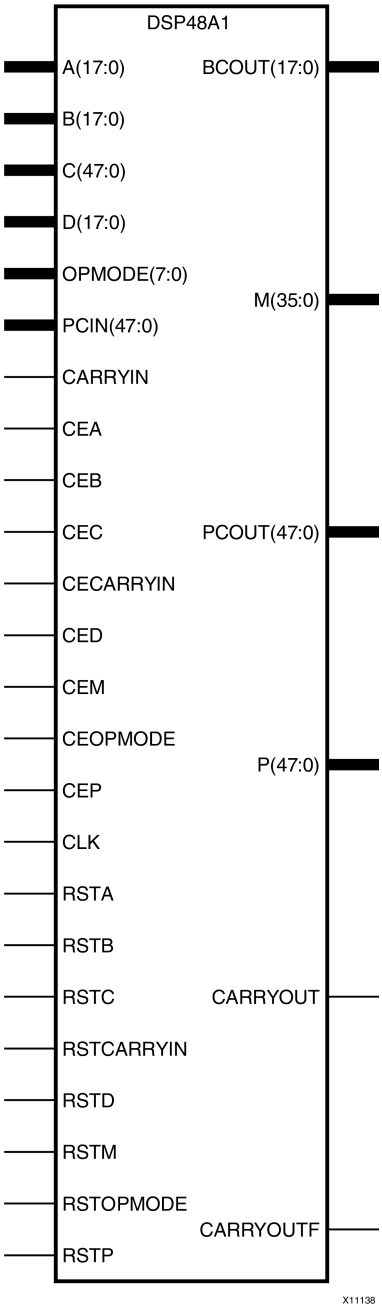
```
// DNA_PORT: Device DNA Data Access Port
//           Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

DNA_PORT #(
   .SIM_DNA_VALUE(57'h000000000000000)  // Specifies the Pre-programmed factory ID value
)
DNA_PORT_inst (
   .DOUT(DOUT),   // 1-bit output: DNA output data
   .CLK(CLK),     // 1-bit input: Clock input
   .DIN(DIN),     // 1-bit input: User data input pin
   .READ(READ),   // 1-bit input: Active high load DNA, active low read input
   .SHIFT(SHIFT)  // 1-bit input: Active high shift enable input
);

// End of DNA_PORT_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA Configuration User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.
- See the *Spartan-6 FPGA User Documentation (User Guides and Data Sheets)*.

DSP48A1

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



Introduction

This element 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. Multiple DSP48A1 blocks can be cascaded to form larger multiplication and addition functions. See the *Spartan-6 FPGA DSP48A1 Slice User Guide (UG389)* for additional details on the use of this element.

Port Descriptions

Port	Direction	Width	Function
A[17:0]	Input	18	18-bit data input to the multiplier or the post add/sub depending on the value of OPMODE[1:0].
B[17:0]	Input	18	18-bit data input to the multiplier, the pre-add/sub, and optionally to the post-add/sub depending on the value of OPMODE[1:0].
BCOUT[17:0]	Output	18	Cascade output for Port B. If used, connect to the B port of downstream cascaded DSP48A1. If not used, leave unconnected.
C[47:0]	Input	48	48-bit data input to post-add/sub.
CARRYIN	Input	1	External carry input to the post-add/sub. Connect only to the CARRYOUT pin of another DSP48A1 block.
CARRYOUT	Output	1	Carry out signal for the post-add/sub. Connect only to the CARRYIN pin of another DSP48A1.
CARRYOUTF	Output	1	Carry out signal for the post-add/sub that can be routed into the fabric.
CEA	Input	1	Active high clock enable for the A port registers (A0REG=1 or A1REG=1). Tie to logic one if not used and A0REG=1 or A1REG=1. Tie to logic zero if A0REG=0 and A1REG=0.
CEB	Input	1	Active high clock enable for the B port registers (B0REG=1 or B1REG=1). Tie to logic one if not used and B0REG=1 or B1REG=1. Tie to logic zero if B0REG=0 and B1REG=0.
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 logic zero if CREG=0.
CECARRYIN	Input	1	Active high, clock enable for the carry-in registers (CARRYINREG=1). Tie to logic one if not used and CARRYINREG=1. Tie to a logic zero if CARRINREG=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 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 logic zero if MREG=0.
CEOPMODE	Input	1	Active high clock enable for the OPMODE input registers (OPMODEREG=1). Tie to logic one if not used and OPMODEREG=1. Tie to logic zero if OPMODEREG=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 logic zero if PREG=0.
CLK	Input	1	DSP48A1 clock
D[17:0]	Input	18	18-bit data input to the pre-add/sub.
M[35:0]	Output	36	Direct multiplier data output to fabric. Do not use if P is used.

Port	Direction	Width	Function
OPMODE	Input	8	Control input to select the arithmetic operations of the DSP48A1. • OPMODE[1:0] - Specifies the source of the X input to the post-add/sub. – 0 - Specifies to place all zeros (disable the post-add/sub). – 1 - Use the multiplier product. – 2 - Use the POUT output signal. – 3 - Use the concatenated D, B, A input signals. • OPMODE[3:2] - Specifies the source of the Z input to the post-add/sub. – 0 - Disable the post-add/sub and propagate the multiplier product to POUT. – 1 - Use PCIN. – 2 - Use the POUT port (accumulator). – 3 - Use the C port. • OPMODE[4] - Specifies the use of the pre-add/sub – 0 - Bypass the pre-adder, supplying the data on Port B directly to the multiplier. – 1 - Use the pre-adder, adding or subtracting the values on the B and D ports prior 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.
P[47:0]	Output	48	Primary data output.
PCIN[47:0]	Input	48	Cascade input for Port P. If used, connect to PCOUT of upstream cascaded DSP48A1. If not used, tie port to all zeros.
PCOUT[47:0]	Output	48	Cascade output for Port P. If used, connect to PCIN of downstream cascaded DSP48A1. If not used, leave unconnected.
RSTA	Input	1	Active high reset for the A port registers (A0REG=1 or A1REG=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.
RSTB	Input	1	Active high reset for the B port registers (B0REG=1 or B1REG=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.
RSTC	Input	1	Active high reset for the C port 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.

Port	Direction	Width	Function
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.
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 of the RSTTYPE attribute.
RSTOPMODE	Input	1	Active high reset for the OPMODE registers (OPMODEREG=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.
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.

Design Entry Method

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

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
A0REG	Integer	0, 1	0	Selects usage of first stage A input pipeline registers. Set to 1 to use the first stage A pipeline registers.
A1REG	Integer	1, 0	1	Selects usage of second stage A input pipeline registers. Set to 1 to use the second stage A pipeline registers.
B0REG	Integer	0, 1	0	Selects usage of first stage B input pipeline registers. Set to 1 to use the first stage B pipeline registers.
B1REG	Integer	1, 0	1	Selects usage of second stage B input pipeline registers. Set to 1 to use the second stage B pipeline registers. The second stage B pipeline registers are after the pre-adder circuit.
CARRYINREG	Integer	1, 0	1	Selects usage of the CARRYIN input pipeline registers. Set to 1 to use the CARRYIN pipeline registers.
CARRYINSEL	String	"CARRYIN", "OPMODE5"	"OPMODE5"	Selects whether the post add/sub carry-in signal should be sourced from the CARRYIN pin (connected to the CARRYOUT of another DSP48A1) or dynamically controlled from the FPGA fabric by the OPMODE[5] input.

Attribute	Data Type	Allowed Values	Default	Description
CARRYOUTREG	Integer	1, 0	1	Selects usage of CARRYOUT output pipeline registers. Set to 1 to use the CARRYOUT pipeline regsiters. The registered outputs will include CARRYOUT and CARRYOUTF.
CREG	Integer	1, 0	1	Selects usage of the C input pipeline registers. Set to 1 to use the C pipeline registers.
DREG	Integer	1, 0	1	Selects usage of D pre-adder input pipeline registers. Set to 1 to use the D pipeline registers.
MREG	Integer	1, 0	1	Selects usage of M multiplier output pipeline registers. Set to 1 to use the M pipeline registers.
OPMODEREG	Integer	1, 0	1	Selects usage of OPMODE input pipeline registers. Set to 1 to use the OPMODE pipeline registers.
PREG	Integer	1, 0	1	Selects usage of P output pipeline registers. Set to 1 to use the P pipeline registers. The registered outputs will include P and PCOUT.
RSTTYPE	String	"SYNC", "ASYNC"	"SYNC"	Selects whether all resets for the DSP48A1 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.

VHDL Instantiation Template

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

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


-- DSP48A1: 48-bit Multi-Functional Arithmetic Block
--          Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

DSP48A1_inst : DSP48A1
generic map (
   A0REG => 0,              -- First stage A input pipeline register (0/1)
   A1REG => 1,              -- Second stage A input pipeline register (0/1)
   B0REG => 0,              -- First stage B input pipeline register (0/1)
   B1REG => 1,              -- Second stage B input pipeline register (0/1)
   CARRYINREG => 1,         -- CARRYIN input pipeline register (0/1)
   CARRYINSEL => "OPMODE5", -- Specify carry-in source, "CARRYIN" or "OPMODE5"
   CARRYOUTREG => 1,        -- CARRYOUT output pipeline register (0/1)
   CREG => 1,               -- C input pipeline register (0/1)
   DREG => 1,               -- D pre-adder input pipeline register (0/1)
   MREG => 1,               -- M pipeline register (0/1)
   OPMODEREG => 1,          -- Enable=1/disable=0 OPMODE input pipeline registers
   PREG => 1,               -- P output pipeline register (0/1)
   RSTTYPE => "SYNC"        -- Specify reset type, "SYNC" or "ASYNC"
)
port map (
   -- Cascade Ports: 18-bit (each) output: Ports to cascade from one DSP48 to another
   BCOUT => BCOUT,           -- 18-bit output: B port cascade output
   PCOUT => PCOUT,           -- 48-bit output: P cascade output (if used, connect to PCIN of another DSP48A1)
   -- Data Ports: 1-bit (each) output: Data input and output ports
   CARRYOUT => CARRYOUT,     -- 1-bit output: carry output (if used, connect to CARRYIN pin of another
                             -- DSP48A1)
```

```
   CARRYOUTF => CARRYOUTF,   -- 1-bit output: fabric carry output
   M => M,                   -- 36-bit output: fabric multiplier data output
   P => P,                   -- 48-bit output: data output
   -- Cascade Ports: 48-bit (each) input: Ports to cascade from one DSP48 to another
   PCIN => PCIN,             -- 48-bit input: P cascade input (if used, connect to PCOUT of another DSP48A1)
   -- Control Input Ports: 1-bit (each) input: Clocking and operation mode
   CLK => CLK,               -- 1-bit input: clock input
   OPMODE => OPMODE,         -- 8-bit input: operation mode input
   -- Data Ports: 18-bit (each) input: Data input and output ports
   A => A,                   -- 18-bit input: A data input
   B => B,                   -- 18-bit input: B data input (connected to fabric or BCOUT of adjacent DSP48A1)
   C => C,                   -- 48-bit input: C data input
   CARRYIN => CARRYIN,       -- 1-bit input: carry input signal (if used, connect to CARRYOUT pin of another
                             -- DSP48A1)

   D => D,                   -- 18-bit input: B pre-adder data input
   -- Reset/Clock Enable Input Ports: 1-bit (each) input: Reset and enable input ports
   CEA => CEA,               -- 1-bit input: active high clock enable input for A registers
   CEB => CEB,               -- 1-bit input: active high clock enable input for B registers
   CEC => CEC,               -- 1-bit input: active high clock enable input for C registers
   CECARRYIN => CECARRYIN,   -- 1-bit input: active high clock enable input for CARRYIN registers
   CED => CED,               -- 1-bit input: active high clock enable input for D registers
   CEM => CEM,               -- 1-bit input: active high clock enable input for multiplier registers
   CEOPMODE => CEOPMODE,     -- 1-bit input: active high clock enable input for OPMODE registers
   CEP => CEP,               -- 1-bit input: active high clock enable input for P registers
   RSTA => RSTA,             -- 1-bit input: reset input for A pipeline registers
   RSTB => RSTB,             -- 1-bit input: reset input for B pipeline registers
   RSTC => RSTC,             -- 1-bit input: reset input for C pipeline registers
   RSTCARRYIN => RSTCARRYIN, -- 1-bit input: reset input for CARRYIN pipeline registers
   RSTD => RSTD,             -- 1-bit input: reset input for D pipeline registers
   RSTM => RSTM,             -- 1-bit input: reset input for M pipeline registers
   RSTOPMODE => RSTOPMODE,   -- 1-bit input: reset input for OPMODE pipeline registers
   RSTP => RSTP              -- 1-bit input: reset input for P pipeline registers
);

-- End of DSP48A1_inst instantiation
```

Verilog Instantiation Template

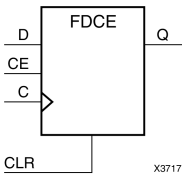
```
// DSP48A1: 48-bit Multi-Functional Arithmetic Block
//          Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

DSP48A1 #(
   .A0REG(0),              // First stage A input pipeline register (0/1)
   .A1REG(1),              // Second stage A input pipeline register (0/1)
   .B0REG(0),              // First stage B input pipeline register (0/1)
   .B1REG(1),              // Second stage B input pipeline register (0/1)
   .CARRYINREG(1),         // CARRYIN input pipeline register (0/1)
   .CARRYINSEL("OPMODE5"), // Specify carry-in source, "CARRYIN" or "OPMODE5"
   .CARRYOUTREG(1),        // CARRYOUT output pipeline register (0/1)
   .CREG(1),               // C input pipeline register (0/1)
   .DREG(1),               // D pre-adder input pipeline register (0/1)
   .MREG(1),               // M pipeline register (0/1)
   .OPMODEREG(1),          // Enable=1/disable=0 OPMODE input pipeline registers
   .PREG(1),               // P output pipeline register (0/1)
   .RSTTYPE("SYNC")        // Specify reset type, "SYNC" or "ASYNC"
)
DSP48A1_inst (
   // Cascade Ports: 18-bit (each) output: Ports to cascade from one DSP48 to another
   .BCOUT(BCOUT),           // 18-bit output: B port cascade output
   .PCOUT(PCOUT),           // 48-bit output: P cascade output (if used, connect to PCIN of another DSP48A1)
   // Data Ports: 1-bit (each) output: Data input and output ports
   .CARRYOUT(CARRYOUT),     // 1-bit output: carry output (if used, connect to CARRYIN pin of another
                            // DSP48A1)

   .CARRYOUTF(CARRYOUTF),   // 1-bit output: fabric carry output
   .M(M),                   // 36-bit output: fabric multiplier data output
   .P(P),                   // 48-bit output: data output
   // Cascade Ports: 48-bit (each) input: Ports to cascade from one DSP48 to another
   .PCIN(PCIN),             // 48-bit input: P cascade input (if used, connect to PCOUT of another DSP48A1)
   // Control Input Ports: 1-bit (each) input: Clocking and operation mode
   .CLK(CLK),               // 1-bit input: clock input
   .OPMODE(OPMODE),         // 8-bit input: operation mode input
   // Data Ports: 18-bit (each) input: Data input and output ports
   .A(A),                   // 18-bit input: A data input
   .B(B),                   // 18-bit input: B data input (connected to fabric or BCOUT of adjacent DSP48A1)
   .C(C),                   // 48-bit input: C data input
   .CARRYIN(CARRYIN),       // 1-bit input: carry input signal (if used, connect to CARRYOUT pin of another
                            // DSP48A1)

   .D(D),                   // 18-bit input: B pre-adder data input
   // Reset/Clock Enable Input Ports: 1-bit (each) input: Reset and enable input ports
   .CEA(CEA),               // 1-bit input: active high clock enable input for A registers
   .CEB(CEB),               // 1-bit input: active high clock enable input for B registers
   .CEC(CEC),               // 1-bit input: active high clock enable input for C registers
   .CECARRYIN(CECARRYIN),   // 1-bit input: active high clock enable input for CARRYIN registers
   .CED(CED),               // 1-bit input: active high clock enable input for D registers
   .CEM(CEM),               // 1-bit input: active high clock enable input for multiplier registers
   .CEOPMODE(CEOPMODE),     // 1-bit input: active high clock enable input for OPMODE registers
   .CEP(CEP),               // 1-bit input: active high clock enable input for P registers
   .RSTA(RSTA),             // 1-bit input: reset input for A pipeline registers
   .RSTB(RSTB),             // 1-bit input: reset input for B pipeline registers
   .RSTC(RSTC),             // 1-bit input: reset input for C pipeline registers
   .RSTCARRYIN(RSTCARRYIN), // 1-bit input: reset input for CARRYIN pipeline registers
   .RSTD(RSTD),             // 1-bit input: reset input for D pipeline registers
   .RSTM(RSTM),             // 1-bit input: reset input for M pipeline registers
   .RSTOPMODE(RSTOPMODE),   // 1-bit input: reset input for OPMODE pipeline registers
   .RSTP(RSTP)              // 1-bit input: reset input for P pipeline registers
);

// End of DSP48A1_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.
- See the *Spartan-6 FPGA DSP48A1 Slice User Guide*.

FDCE

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



Introduction

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

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

Logic Table

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

Design Entry Method

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

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
INIT	Binary	0, 1	0	Sets the initial value of Q output after configuration For Spartan®-6, the INIT value should always match the polarity of the set or reset. For this element, the INIT should be 0. If set to 1, an asynchronous circuit must be created to exhibit this behavior, which Xilinx does not recommend.

VHDL Instantiation Template

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

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


-- FDCE: Single Data Rate D Flip-Flop with Asynchronous Clear and
--       Clock Enable (posedge clk).
--       Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of FDCE_inst instantiation
```

Verilog Instantiation Template

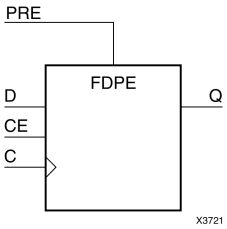
```
// FDCE: Single Data Rate D Flip-Flop with Asynchronous Clear and
//       Clock Enable (posedge clk).
//       Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

FDCE FDCE_inst (
   .Q(Q),      // 1-bit Data output
   .C(C),      // 1-bit Clock input
   .CE(CE),    // 1-bit Clock enable input
   .CLR(CLR),  // 1-bit Asynchronous clear input
   .D(D)       // 1-bit Data input
);

// End of FDCE_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA Configurable Logic Block User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

FDPE

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



Introduction

This design element is a single D-type flip-flop with data (D), clock enable (CE), and asynchronous preset (PRE) inputs and data output (Q). The asynchronous PRE, when High, overrides all other inputs and sets the (Q) output High. Data on the (D) input is loaded into the flip-flop when PRE is Low and CE is High on the Low-to-High clock (C) transition. When CE is Low, the clock transitions are ignored.

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

Logic Table

Inputs				Outputs
PRE	**CE**	**D**	**C**	**Q**
1	X	X	X	1
0	0	X	X	No Change
0	1	D	↑	D

Design Entry Method

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

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
INIT	Binary	0, 1	1	Sets the initial value of Q output after configuration For Spartan®-6, Xilinx recommends that the INIT value always matches the polarity of the set or reset. For this element, the INIT should be 1. If set to 0, additional asynchronous circuitry will be created to correctly model the behavior.

VHDL Instantiation Template

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

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


-- FDPE: Single Data Rate D Flip-Flop with Asynchronous Preset and
--       Clock Enable (posedge clk).
--       Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

FDPE_inst : FDPE
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
   PRE => PRE,   -- Asynchronous preset input
   D => D        -- Data input
);

-- End of FDPE_inst instantiation
```

Verilog Instantiation Template

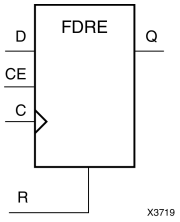
```
// FDPE: Single Data Rate D Flip-Flop with Asynchronous Preset and
//       Clock Enable (posedge clk).
//       Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

FDPE FDPE_inst (
   .Q(Q),      // 1-bit Data output
   .C(C),      // 1-bit Clock input
   .CE(CE),    // 1-bit Clock enable input
   .PRE(PRE),  // 1-bit Asynchronous preset input
   .D(D)       // 1-bit Data input
);

// End of FDPE_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA Configurable Logic Block User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

FDRE

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



Introduction

This design element is a single D-type flip-flop with data (D), clock enable (CE), and synchronous reset (R) inputs and data output (Q). The synchronous reset (R) input, when High, overrides all other inputs and resets the (Q) output Low on the Low-to-High clock (C) transition. The data on the (D) input is loaded into the flip-flop when R is Low and CE is High during the Low-to-High 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	**CE**	**D**	**C**	**Q**
1	X	X	↑	0
0	0	X	X	No Change
0	1	D	↑	D

Design Entry Method

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

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
INIT	Binary	0, 1	0	Sets the initial value of Q output after configuration For Spartan®-6, the INIT value should always match the polarity of the set or reset. For this element, the INIT should be 0. If set to 1, an asynchronous circuit must be created to exhibit this behavior, which Xilinx does not recommend.

VHDL Instantiation Template

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

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


-- FDRE: Single Data Rate D Flip-Flop with Synchronous Reset and
--       Clock Enable (posedge clk).
--       Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

FDRE_inst : FDRE
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
   R => R,      -- Synchronous reset input
   D => D       -- Data input
);

-- End of FDRE_inst instantiation
```

Verilog Instantiation Template

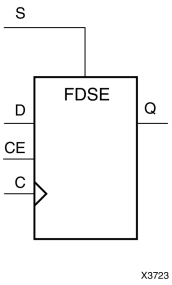
```
// FDRE: Single Data Rate D Flip-Flop with Synchronous Reset and
//       Clock Enable (posedge clk).
//       Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

FDRE FDRE_inst (
   .Q(Q),      // 1-bit Data output
   .C(C),      // 1-bit Clock input
   .CE(CE),    // 1-bit Clock enable input
   .R(R),      // 1-bit Synchronous reset input
   .D(D)       // 1-bit Data input
);

// End of FDRE_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA Configurable Logic Block User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

FDSE

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



Introduction

FDSE is a single D-type flip-flop with data (D), clock enable (CE), and synchronous set (S) inputs and data output (Q). The synchronous set (S) input, when High, overrides the clock enable (CE) input and sets the Q output High during the Low-to-High clock (C) transition. The data on the D input is loaded into the flip-flop when S is Low and CE is High during the Low-to-High clock (C) transition.

For FPGA devices, this flip-flop is asynchronously preset, output High, when power is applied. 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
S	**CE**	**D**	**C**	**Q**
1	X	X	↑	1
0	0	X	X	No Change
0	1	D	↑	D

Design Entry Method

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

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
INIT	Binary	0, 1	1	Sets the initial value of Q output after configuration For Spartan®-6, Xilinx recommends that the INIT value always matches the polarity of the set or reset. For this element, the INIT should be 1. If set to 0, additional asynchronous circuitry will be created to correctly model the behavior.

VHDL Instantiation Template

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

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


-- FDSE: Single Data Rate D Flip-Flop with Synchronous Set and
--       Clock Enable (posedge clk).
--       Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

FDSE_inst : FDSE
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
   S => S,      -- Synchronous Set input
   D => D       -- Data input
);

-- End of FDSE_inst instantiation
```

Verilog Instantiation Template

```
// FDSE: Single Data Rate D Flip-Flop with Synchronous Set and
//       Clock Enable (posedge clk).
//       Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

FDSE FDSE_inst (
   .Q(Q),      // 1-bit Data output
   .C(C),      // 1-bit Clock input
   .CE(CE),    // 1-bit Clock enable input
   .S(S),      // 1-bit Synchronous set input
   .D(D)       // 1-bit Data input
);

// End of FDSE_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA Configurable Logic Block User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

GTPA1_DUAL

Primitive: Dual Gigabit Transceiver

GTPA1_DUAL

X11154

Introduction

This design element represents the Spartan®-6 FPGA RocketIO™ GTP transceiver, a power-efficient and highly configurable transceiver. Refer to *Spartan-6 FPGA RocketIO GTP Transceiver User Guide* for detailed information regarding this component. The Spartan-6 FPGA RocketIO GTX Transceiver Wizard is the preferred tool to generate a wrapper to instantiate a GTPA1_DUAL primitive. The Wizard can be found in the Xilinx® CORE Generator™ tool.

Design Entry Method

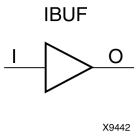
To instantiate this component, use the Spartan-6 FPGA RocketIO GTX Transceiver Wizard or an associated core containing the component. Xilinx does not recommend direct instantiation of this component.

For More Information

- See the *Spartan-6 FPGA RocketIO GTP Transceivers User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

IBUF

Primitive: Input Buffer



Introduction

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

Port Descriptions

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

Design Entry Method

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

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

Available Attributes

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

VHDL Instantiation Template

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

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


-- IBUF: Single-ended Input Buffer
--       Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

IBUF_inst : IBUF
generic map (
   IBUF_LOW_PWR => TRUE, -- Low power (TRUE) vs. performance (FALSE) setting for referenced I/O standards
   IOSTANDARD => "DEFAULT")
port map (
   O => O,     -- Buffer output
   I => I      -- Buffer input (connect directly to top-level port)
);

-- End of IBUF_inst instantiation
```

Verilog Instantiation Template

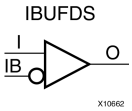
```
// IBUF: Single-ended Input Buffer
//       Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

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

// End of IBUF_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA SelectIO Resources User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

IBUFDS

Primitive: Differential Signaling Input Buffer



Introduction

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

Logic Table

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

Port Descriptions

Port	Direction	Width	Function
I	Input	1	Diff_p Buffer Input
IB	Input	1	Diff_n Buffer Input
O	Output	1	Buffer Output

Design Entry Method

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

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

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
DIFF_TERM	Boolean	TRUE or FALSE	FALSE	Enables the built-in differential termination resistor.
IOSTANDARD	String	See Data Sheet.	"DEFAULT"	Assigns an I/O standard to the element.

VHDL Instantiation Template

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

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


-- IBUFDS: Differential Input Buffer
--         Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

IBUFDS_inst : IBUFDS
generic map (
   DIFF_TERM => FALSE, -- Differential Termination
   IBUF_LOW_PWR => TRUE, -- Low power (TRUE) vs. performance (FALSE) setting for referenced I/O standards
   IOSTANDARD => "DEFAULT")
port map (
   O => O,  -- Buffer output
   I => I,  -- Diff_p buffer input (connect directly to top-level port)
   IB => IB -- Diff_n buffer input (connect directly to top-level port)
);

-- End of IBUFDS_inst instantiation
```

Verilog Instantiation Template

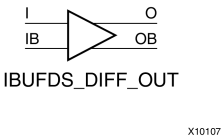
```
// IBUFDS: Differential Input Buffer
//         Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

IBUFDS #(
   .DIFF_TERM("FALSE"),   // Differential Termination
   .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-6 FPGA SelectIO Resources User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

IBUFDS_DIFF_OUT

Primitive: Signaling Input Buffer with Differential Output



Introduction

This design element is an input buffer that supports differential signaling. In IBUFDS_DIFF_OUT, 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). The IBUFDS_DIFF_OUT differs from the IBUFDS in that it allows internal access to both phases of the differential signal. Optionally, a programmable differential termination feature is available to help improve signal integrity and reduce external components.

Logic Table

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

Design Entry Method

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

It is suggested to 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 and OB ports to the logic in which this input is to source. Specify the desired generic/parameter values in order to configure the proper behavior of the buffer.

Available Attributes

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

VHDL Instantiation Template

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

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


-- IBUFDS_DIFF_OUT: Differential Input Buffer with Differential Output
--                  Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

IBUFDS_DIFF_OUT_inst : IBUFDS_DIFF_OUT
generic map (
   DIFF_TERM => FALSE, -- Differential Termination
   IOSTANDARD => "DEFAULT") -- Specify the input I/O standard
port map (
   O => O,     -- Buffer diff_p output
   OB => OB,   -- Buffer diff_n 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_DIFF_OUT_inst instantiation
```

Verilog Instantiation Template

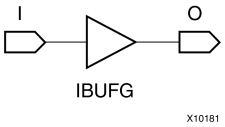
```
// IBUFDS_DIFF_OUT: Differential Input Buffer with Differential Output
//                  Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

IBUFDS_DIFF_OUT #(
   .DIFF_TERM("FALSE"),   // Differential Termination, "TRUE"/"FALSE"
   .IOSTANDARD("DEFAULT") // Specify the input I/O standard
) IBUFDS_DIFF_OUT_inst (
   .O(O),   // Buffer diff_p output
   .OB(OB), // Buffer diff_n 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_DIFF_OUT_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA Clocking Resources User Guide*.
- See the *Spartan-6 FPGA SelectIO Resources User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

IBUFG

Primitive: Dedicated Input Clock Buffer



Introduction

The IBUFG is a dedicated input to the device which should be used to connect incoming clocks to the FPGA's global clock routing resources. The IBUFG provides dedicated connections to the DCM, PLL, or BUFG resources. providing the minimum amount of clock delay and jitter to the device. The IBUFG input can only be driven by the global clock (GC) pins.

Port Descriptions

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

Design Entry Method

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

Available Attributes

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

VHDL Instantiation Template

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

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


-- IBUFG: Single-ended global clock input buffer
--        Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

IBUFG_inst : IBUFG
generic map (
   IBUF_LOW_PWR => TRUE, -- Low power (TRUE) vs. performance (FALSE) setting for referenced I/O standards
   IOSTANDARD => "DEFAULT")
port map (
   O => O, -- Clock buffer output
   I => I  -- Clock buffer input (connect directly to top-level port)
);
```

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

Verilog Instantiation Template

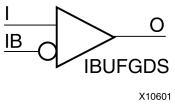
```
// IBUFG: Single-ended global clock input buffer
//        Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

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

// End of IBUFG_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA SelectIO Resources User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

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

Logic Table

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

Port Descriptions

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

Design Entry Method

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

Put all I/O components on the top-level of the design to help facilitate hierarchical design methods. Connect the I port directly to the top-level "master" input port of the design, the IB port to the top-level "slave" input port and the O port to an MMCM, 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	Data Type	Allowed Values	Default	Description
IOSTANDARD	String	See Data Sheet	"DEFAULT"	Assigns an I/O standard to the element.

VHDL Instantiation Template

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

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


-- IBUFGDS: Differential Global Clock Input Buffer
--          Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

IBUFGDS_inst : IBUFGDS
generic map (
   DIFF_TERM => FALSE, -- Differential Termination
   IBUF_LOW_PWR => TRUE, -- Low power (TRUE) vs. performance (FALSE) setting for referenced I/O standards
   IOSTANDARD => "DEFAULT")
port map (
   O => O,  -- Clock buffer output
   I => I,  -- Diff_p clock buffer input (connect directly to top-level port)
   IB => IB -- Diff_n clock buffer input (connect directly to top-level port)
);

-- End of IBUFGDS_inst instantiation
```

Verilog Instantiation Template

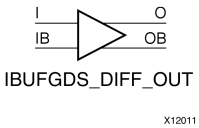
```
// IBUFGDS: Differential Global Clock Input Buffer
//          Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

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

// End of IBUFGDS_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA SelectIO Resources User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

IBUFGDS_DIFF_OUT

Primitive: Differential Signaling Input Buffer with Differential Output



Introduction

This design element is an input buffer that supports differential signaling. In IBUFGDS_DIFF_OUT, 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). The IBUFGDS_DIFF_OUT differs from the IBUFGDS in that it allows internal access to both phases of the differential signal. Optionally, a programmable differential termination feature is available to help improve signal integrity and reduce external components.

Logic Table

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

Port Descriptions

Port	Direction	Width	Function
I	Input	1	Diff_p Buffer Input (connect to top-level port in the design).
IB	Input	1	Diff_n Buffer Input (connect to top-level port in the design).
O	Output	1	Diff_p Buffer Output.
OB	Output	1	Diff_n Buffer Output.

Design Entry Method

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

It is suggested to 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 and OB ports to the logic in which this input is to source. Specify the desired generic/parameter values in order to configure the proper behavior of the buffer.

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
IOSTANDARD	String	See Data Sheet	"DEFAULT"	Assigns an I/O standard to the element.
DIFF_TERM	Boolean	TRUE, FALSE	FALSE	Specifies the use of the internal differential termination resistance.

VHDL 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_DIFF_OUT: Differential Global Clock Buffer with Differential Output
--                   Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

IBUFGDS_DIFF_OUT_inst : IBUFGDS_DIFF_OUT
generic map (
   DIFF_TERM => FALSE, -- Differential Termination
   IOSTANDARD => "DEFAULT") -- Specify the input I/O standard
port map (
   O => O,     -- Buffer diff_p output
   OB => OB,   -- Buffer diff_n 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 IBUFGDS_DIFF_OUT_inst instantiation
```

Verilog Instantiation Template

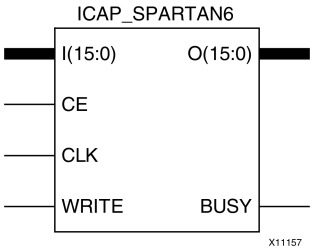
```
// IBUFGDS_DIFF_OUT: Differential Global Clock Buffer with Differential Output
//                   Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

IBUFGDS_DIFF_OUT #(
   .DIFF_TERM("FALSE"),   // Differential Termination, "TRUE"/"FALSE"
   .IOSTANDARD("DEFAULT") // Specify the input I/O standard
) IBUFGDS_DIFF_OUT_inst (
   .O(O),   // Buffer diff_p output
   .OB(OB), // Buffer diff_n 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 IBUFGDS_DIFF_OUT_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA SelectIO Resources User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

ICAP_SPARTAN6

Primitive: Internal Configuration Access Port



Introduction

This design element gives you access to the configuration functions of the FPGA from the FPGA fabric. Using this component, commands and data can be written to and read from the configuration logic of the FPGA array. Since the improper use of this function can have a negative effect on the functionality and reliability of the FPGA, you should not use this element unless you are very familiar with its capabilities.

Port Descriptions

Port	Direction	Width	Function
BUSY	Output	1	Busy/Ready output.
CE	Input	1	Active-Low ICAP Enable input.
CLK	Input	1	Clock input.
I[15:0]	Input	16	Configuration data input bus.
O[15:0]	Output	16	Configuration data output bus.
WRITE	Input	1	Read/Write control input.

Design Entry Method

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

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
DEVICE_ID	Hexadecimal	32'h02000093, 32'h0200E093, 32'h0201D093, 32'h0202E093, 32'h0203D093, 32'h02001093, 32'h02002093, 32'h02004093, 32'h02008093, 32'h02011093, 32'h02024093,	32'h02000093	Specifies the pre-programmed Device ID value.

Attribute	Data Type	Allowed Values	Default	Description
		32'h02028093, 32'h02031093		
SIM_CFG_FILE_NAME	String	String representing file name and location	None	Specifies the Raw Bitstream (RBT) file to be parsed by the simulation model.

VHDL 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_SPARTAN6: Internal Configuration Access Port
--                Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

ICAP_SPARTAN6_inst : ICAP_SPARTAN6
generic map (
   DEVICE_ID => X"4000093",     -- Specifies the pre-programmed Device ID value
   SIM_CFG_FILE_NAME => "NONE"  -- Specifies the Raw Bitstream (RBT) file to be parsed by the simulation
                                -- model
)
port map (
   BUSY => BUSY,   -- 1-bit output: Busy/Ready output
   O => O,         -- 16-bit output: Configuartion data output bus
   CE => CE,       -- 1-bit input: Active-Low ICAP Enable input
   CLK => CLK,     -- 1-bit input: Clock input
   I => I,         -- 16-bit input: Configuration data input bus
   WRITE => WRITE  -- 1-bit input: Read/Write control input
);

-- End of ICAP_SPARTAN6_inst instantiation
```

Verilog Instantiation Template

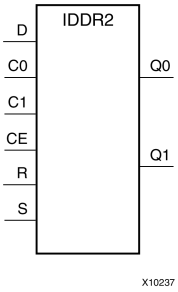
```
// ICAP_SPARTAN6: Internal Configuration Access Port
//                Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

ICAP_SPARTAN6 #(
   .DEVICE_ID(0'h4000093),     // Specifies the pre-programmed Device ID value
   .SIM_CFG_FILE_NAME("NONE")  // Specifies the Raw Bitstream (RBT) file to be parsed by the simulation
                               // model
)
ICAP_SPARTAN6_inst (
   .BUSY(BUSY),   // 1-bit output: Busy/Ready output
   .O(O),         // 16-bit output: Configuartion data output bus
   .CE(CE),       // 1-bit input: Active-Low ICAP Enable input
   .CLK(CLK),     // 1-bit input: Clock input
   .I(I),         // 16-bit input: Configuration data input bus
   .WRITE(WRITE)  // 1-bit input: Read/Write control input
);

// End of ICAP_SPARTAN6_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA Configuration User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

IDDR2

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



Introduction

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

Logic Table

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

Design Entry Method

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

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

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
DDR_ALIGNMENT	String	"NONE", "C0", "C1"	"NONE"	Sets the output alignment more for the DDR register • NONE - Makes the data available on the Q0 and Q1 outputs shortly after the corresponding C0 or C1 positive clock edge. • C0 - Makes the data on both Q0 and Q1 align to the positive edge of the C0 clock. • C1 - Makes the data on both Q0 and Q1 align to the positive edge of the C1 clock.
INIT_Q0	Integer	0, 1	0	Sets initial state of the Q0 output to 0 or 1.
INIT_Q1	Integer	0, 1	0	Sets initial state of the Q1 output to 0 or 1.
SRTYPE	String	"SYNC", "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-6
-- Xilinx HDL Libraries Guide, version 14.1

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

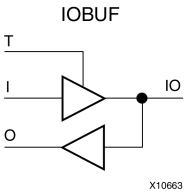
```
// IDDR2: Input Double Data Rate Input Register with Set, Reset
//        and Clock Enable.
//        Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

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-6 FPGA Configurable Logic Block User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

IOBUF

Primitive: Bi-Directional Buffer



Introduction

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

Logic Table

Inputs		Bidirectional	Outputs
T	**I**	**IO**	**O**
1	X	Z	IO
0	1	1	1
0	0	0	0

Port Descriptions

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

Design Entry Method

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

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
DRIVE	Integer	2, 4, 6, 8, 12, 16, 24	12	Selects output drive strength (mA) for the SelectIO™ buffers that use the LVTTL, LVCMOS12, LVCMOS15, LVCMOS18, LVCMOS25, or LVCMOS33 interface I/O standard.
IOSTANDARD	String	See Data Sheet	"DEFAULT"	Assigns an I/O standard to the element.
SLEW	String	"SLOW", "FAST", "QUIETIO"	"SLOW"	Sets the output rise and fall time. See the Data Sheet for recommendations of the best setting for this attribute.

VHDL Instantiation Template

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

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


-- IOBUF: Single-ended Bi-directional Buffer
--        Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of IOBUF_inst instantiation
```

Verilog Instantiation Template

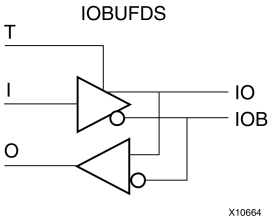
```
// IOBUF: Single-ended Bi-directional Buffer
//        Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

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

// End of IOBUF_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA SelectIO Resources User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

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

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

Design Entry Method

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

Available Attributes

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

VHDL Instantiation Template

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

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


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

IOBUFDS_inst : IOBUFDS
generic map (
   IOSTANDARD => "BLVDS_25")
port map (
   O => O,     -- Buffer output
   IO => IO,   -- Diff_p inout (connect directly to top-level port)
   IOB => IOB, -- Diff_n inout (connect directly to top-level port)
   I => I,     -- Buffer input
   T => T      -- 3-state enable input, high=input, low=output
);

-- End of IOBUFDS_inst instantiation
```

Verilog Instantiation Template

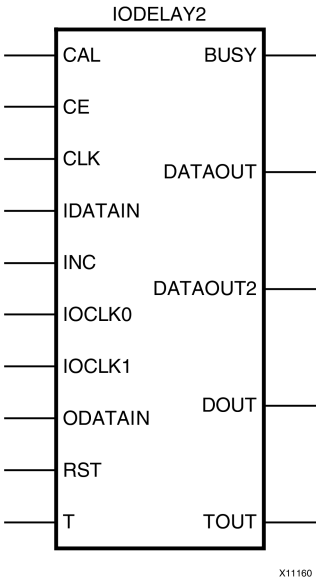
```
// IOBUFDS: Differential Bi-directional Buffer
//          Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

IOBUFDS #(
   .IOSTANDARD("BLVDS_25")    // Specify the I/O standard
) IOBUFDS_inst (
   .O(O),     // Buffer output
   .IO(IO),   // Diff_p inout (connect directly to top-level port)
   .IOB(IOB), // Diff_n inout (connect directly to top-level port)
   .I(I),     // Buffer input
   .T(T)      // 3-state enable input, high=input, low=output
);

// End of IOBUFDS_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA SelectIO Resources User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

IODELAY2

Primitive: Input and Output Fixed or Variable Delay Element



Introduction

This design element can be used to provide a fixed delay or an adjustable delay to the input path and a fixed delay for the output path of the Spartan®-6 FPGA. This delay can be useful for data alignment of incoming or outgoing data to/from the chip. When used in variable mode, the input path can be adjusted for increasing and decreasing amounts of delay. The output delay path is only available in a fixed delay. The IODELAY can also be used to add additional static or variable delay to an internal path (within the FPGA fabric). However, when IODELAY is used that way, this device is no longer available to the associated I/O for input or output path delays.

Port Descriptions

Port	Direction	Width	Function
BUSY	Output	1	Busy after CAL.
CAL	Input	1	Initiate calibration input.
CE	Input	1	Enable increment/decrement.
CLK	Input	1	IODELAY Clock input.
DATAOUT	Output	1	Delayed Data output from input port (connect to input datapath logic, can only route to a register in ILOGIC).
DATAOUT2	Output	1	Delayed Data output from input port (connect to input datapath logic, can route to fabric).
DOUT	Output	1	Delayed Data Output to IOB.
IDATAIN	Input	1	Data Signal from IOB.
INC	Input	1	Increment / Decrement Input.
IOCLK0	Input	1	Optionally Invertible I/O clock inputs.
IOCLK1	Input	1	Optionally Invertible I/O clock inputs.

Port	Direction	Width	Function
ODATAIN	Input	1	Output Data Input from OLOGIC or OSERDES.
RST	Input	1	Reset the IODELAY2 to either zero or 1/2 of total period. RST_VALUE attribute controls this choice.
T	Input	1	3-state input control. Tie high for input-only or internal delay or tie low for output only.
TOUT	Output	1	Delayed 3-state signal output. Tie high for input-only or internal delay or tie low for output only.

Design Entry Method

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

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
COUNTER_WRAPAROUND	String	"WRAPAROUND", STAY_AT_LIMIT"	"WRAPAROUND"	Sets behavior when tap count exceeds max or min, depending on whether tap setting is being incremented or decremented.
DATA_RATE	String	"SDR", "DDR"	"SDR"	Single Data Rate or Double Data Rate operation.
DELAY_SRC	String	"IO", "IDATAIN", "ODATAIN"	"IO"	• ODATAIN indicates delay source is the ODATAIN pin from the OSERDES or OLOGIC. • IDATAIN indicates the delay source is from the IDATAIN pin; one of the dedicated IOB (P/N) Pads. • IO means that the signal source switches between IDATAIN and ODATAIN depending on the sense of the T (tristate) input.
IDELAY_MODE	String	"NORMAL", "PCI"	"NORMAL"	Do not specify or modify this attribute.

Attribute	Data Type	Allowed Values	Default	Description
IDELAY_TYPE	String	"DEFAULT", "DIFF_PHASE_DETECTOR", "FIXED", "VARIABLE_FROM_HALF_MAX", "VARIABLE_FROM_ZERO"	"DEFAULT"	Delay Type. VARIABLE refers to the customer calibrated delay mode. • DEFAULT utilizes physical chip settings for best approximation of zero hold time programming. • VARIABLE_FROM_ZERO and VARIABLE_FROM_HALF_MAX refer to the reset behavior. • DIFF_PHASE_DETECTOR is a special mode where the master and slave IODELAY2s are cascaded.
IDELAY_VALUE	Integer	0 to 255	0	Delay tap value for IDELAY Mode.
IDELAY2_VALUE	Integer	0 to 255	0	Delay tap value for IDELAY Mode. Only used when IDELAY_MODE is set to PCI.
ODELAY_VALUE	Integer	0 to 255	0	Delay tap value for ODELAY Mode.
SERDES_MODE	String	"NONE", "MASTER", "SLAVE"	"NONE"	Specify whether the ISERDES2 is operating in master or slave modes when cascaded width expansion.
SIM_TAPDELAY_VALUE	Integer	10 to 90	75	A simulation only attribute. Allows setting the nominal tap delay to different settings for simulation.

VHDL Instantiation Template

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

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


-- IODELAY2: Input and Output Fixed or Variable Delay Element
--           Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

IODELAY2_inst : IODELAY2
generic map (
   COUNTER_WRAPAROUND => "WRAPAROUND", -- "STAY_AT_LIMIT" or "WRAPAROUND"
   DATA_RATE => "SDR",                 -- "SDR" or "DDR"
   DELAY_SRC => "IO",                  -- "IO", "ODATAIN" or "IDATAIN"
   IDELAY2_VALUE => 0,                 -- Delay value when IDELAY_MODE="PCI" (0-255)
   IDELAY_MODE => "NORMAL",            -- "NORMAL" or "PCI"
   IDELAY_TYPE => "DEFAULT",           -- "FIXED", "DEFAULT", "VARIABLE_FROM_ZERO", "VARIABLE_FROM_HALF_MAX"
                                       -- or "DIFF_PHASE_DETECTOR"
   IDELAY_VALUE => 0,                  -- Amount of taps for fixed input delay (0-255)
   ODELAY_VALUE => 0,                  -- Amount of taps fixed output delay (0-255)
   SERDES_MODE => "NONE",              -- "NONE", "MASTER" or "SLAVE"
   SIM_TAPDELAY_VALUE => 75            -- Per tap delay used for simulation in ps
)
port map (
   BUSY => BUSY,         -- 1-bit output: Busy output after CAL
```

```
   DATAOUT => DATAOUT,   -- 1-bit output: Delayed data output to ISERDES/input register
   DATAOUT2 => DATAOUT2, -- 1-bit output: Delayed data output to general FPGA fabric
   DOUT => DOUT,         -- 1-bit output: Delayed data output
   TOUT => TOUT,         -- 1-bit output: Delayed 3-state output
   CAL => CAL,           -- 1-bit input: Initiate calibration input
   CE => CE,             -- 1-bit input: Enable INC input
   CLK => CLK,           -- 1-bit input: Clock input
   IDATAIN => IDATAIN,   -- 1-bit input: Data input (connect to top-level port or I/O buffer)
   INC => INC,           -- 1-bit input: Increment / decrement input
   IOCLK0 => IOCLK0,     -- 1-bit input: Input from the I/O clock network
   IOCLK1 => IOCLK1,     -- 1-bit input: Input from the I/O clock network
   ODATAIN => ODATAIN,   -- 1-bit input: Output data input from output register or OSERDES2.
   RST => RST,           -- 1-bit input: Reset to zero or 1/2 of total delay period
   T => T                -- 1-bit input: 3-state input signal
);

-- End of IODELAY2_inst instantiation
```

Verilog Instantiation Template

```
// IODELAY2: Input and Output Fixed or Variable Delay Element
//           Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

IODELAY2 #(
   .COUNTER_WRAPAROUND("WRAPAROUND"), // "STAY_AT_LIMIT" or "WRAPAROUND"
   .DATA_RATE("SDR"),                 // "SDR" or "DDR"
   .DELAY_SRC("IO"),                  // "IO", "ODATAIN" or "IDATAIN"
   .IDELAY2_VALUE(0),                 // Delay value when IDELAY_MODE="PCI" (0-255)
   .IDELAY_MODE("NORMAL"),            // "NORMAL" or "PCI"
   .IDELAY_TYPE("DEFAULT"),           // "FIXED", "DEFAULT", "VARIABLE_FROM_ZERO", "VARIABLE_FROM_HALF_MAX"
                                      // or "DIFF_PHASE_DETECTOR"
   .IDELAY_VALUE(0),                  // Amount of taps for fixed input delay (0-255)
   .ODELAY_VALUE(0),                  // Amount of taps fixed output delay (0-255)
   .SERDES_MODE("NONE"),              // "NONE", "MASTER" or "SLAVE"
   .SIM_TAPDELAY_VALUE(75)            // Per tap delay used for simulation in ps
)
IODELAY2_inst (
   .BUSY(BUSY),         // 1-bit output: Busy output after CAL
   .DATAOUT(DATAOUT),   // 1-bit output: Delayed data output to ISERDES/input register
   .DATAOUT2(DATAOUT2), // 1-bit output: Delayed data output to general FPGA fabric
   .DOUT(DOUT),         // 1-bit output: Delayed data output
   .TOUT(TOUT),         // 1-bit output: Delayed 3-state output
   .CAL(CAL),           // 1-bit input: Initiate calibration input
   .CE(CE),             // 1-bit input: Enable INC input
   .CLK(CLK),           // 1-bit input: Clock input
   .IDATAIN(IDATAIN),   // 1-bit input: Data input (connect to top-level port or I/O buffer)
   .INC(INC),           // 1-bit input: Increment / decrement input
   .IOCLK0(IOCLK0),     // 1-bit input: Input from the I/O clock network
   .IOCLK1(IOCLK1),     // 1-bit input: Input from the I/O clock network
   .ODATAIN(ODATAIN),   // 1-bit input: Output data input from output register or OSERDES2.
   .RST(RST),           // 1-bit input: Reset to zero or 1/2 of total delay period
   .T(T)                // 1-bit input: 3-state input signal
);

// End of IODELAY2_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA SelectIO Resources User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

IODRP2

Primitive: I/O Control Port

Introduction

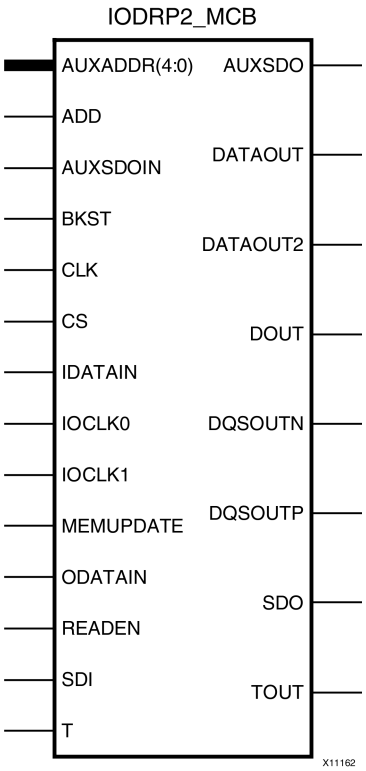
Xilinx does not support the use of this element.

For More Information

- See the *Spartan-6 FPGA SelectIO Resources User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

IODRP2_MCB

Primitive: I/O Control Port for the Memory Controller Block



Introduction

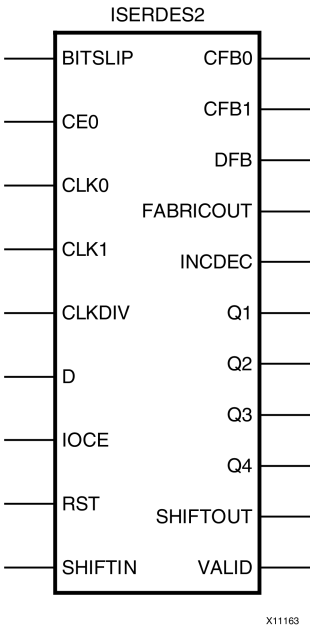
The IODRP2_MCB is a component used by the Memory Interface Generator (MIG) core in conjunction with the MCB block to implement external memory interfaces. The use of this block outside of MIG is not supported.

For More Information

- See the *Xilinx Memory Interface Generator (MIG) User Guide*
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

ISERDES2

Primitive: Input SERial/DESerializer.



Introduction

Each IOB contains an input deserializer block that can be instantiated in a design by using the ISERDES2 primitive. ISERDES2 allows serial-to-parallel conversion with SerDes ratios of 1:2, 1:3 and 1:4. The SerDes ratio is the ratio between the high speed I/O clock that is capturing data, and the slower internal global clock used for processing the parallel data. For example, with a single-rate I/O clock running at 500 MHz to receive data at 500 Mb/s,the ISERDES2 transfers four bits of data at one quarter of the rate (125 MHz) to the FPGA logic. When using differential inputs, the two ISERDES2 primitives associated with the two IOBs can be cascaded to allow higher SerDes ratios of 1:5, 1:6, 1:7, and 1:8. Each ISERDES2 also contains logic to word-align the parallel data, as required by the designer, by performing a Bitslip operation.

Port Descriptions

Port	Direction	Width	Function
BITSLIP	Input	1	Invoke Bitslip when High. Bitslip operation can be used with any DATA_WIDTH, cascaded or not. The amount of Bitslip is fixed by the DATA_WIDTH selection.
CE0	Input	1	Clock enable input for final (global clock driven) register.
CFB0	Output	1	Feed-through route to allow a PLL or DCM generated clock to feed back to the PLL or DCM through a BUFIO2FB.
CFB1	Output	1	Secondary feed-through route to allow a PLL or DCM generated clock to feed back to the PLL or DCM through a BUFIO2FB.
CLKDIV	Input	1	Global clock network input. This is the clock for the FPGA logic domain.
CLK0	Input	1	I/O clock network input. Optionally invertible. This is the primary clock input used when the clock doubler circuit is not engaged.(see DATA_RATE attribute).

Port	Direction	Width	Function
CLK1	Input	1	I/O clock network input. Optionally invertible. This secondary clock input is only used when the clock doubler is engaged (see DATA_RATE attribute).
D	Input	1	Input data. This is the data input after being delayed by the IODELAY2 block.
DFB	Output	1	Feed-through route to allow an input clock that has been delayed in an IODELAY2 element to be forwarded to a DCM, PLL, or BUFG through a BUFIO2.
FABRICOUT	Output	1	Asynchronous data for use in the FPGA logic.
INCDEC	Output	1	Output of phase detector in master mode (dummy in slave mode). Indicates to the FPGA logic whether the received data was sampled early or late.
IOCE	Input	1	Data strobe signal derived from BUFIO CE. Strobes data capture to be correctly timed with respect to the I/O and global clocks for the SerDes mode selected.
Q1 - Q4	Output	1	Registered output to fabric.
RST	Input	1	Asynchronous reset only.
SHIFTIN	Input	1	Cascade-in signal for master/slave I/O. Used when master and slave sites are used together for DATA_WIDTHs greater than four. When the block is a master, it transfers data in for use in the phase-detector mode. When the block is a slave, it transfers serial data in to become parallel data.
SHIFTOUT	Output	1	Cascade-out signal for master/slave I/O. In slave mode, it is used to send sampled data from the slave. In master mode, it sends serial data from the fourth stage of the input shift register to the slave.
VALID	Output	1	Output of the phase detector in master mode (dummy in slave mode). If the input data contains no edges (no information for the phase detector to work with) the VALID signal transitions Low to indicate that the FPGA logic should ignore the INCDEC signal.

Design Entry Method

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

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
BITSLIP_ENABLE	Boolean	FALSE, TRUE	FALSE	Enables or disables the Bitslip function controlled by the BITSLIP input pin. The number of bits slipped is a function of the DATA_WIDTH selected. When disabled, the Bitslip CE always remains at the default value of one I/O clock before the IOCE clock enable.
DATA_RATE	String	"SDR", "DDR"	"SDR"	Data rate setting. The DDR clock can be supplied by separate I/O clocks or by a single I/O clock. If two clocks are supplied, they must be approximately 180 degrees out of phase.
DATA_WIDTH	Integer	1, 2, 3, 4, 5, 6, 7, 8	1	Data width. Defines the parallel data output width of the serial-to-parallel converter. Values greater than four are only valid when two ISERDES2 blocks are cascaded. In this case, the same value should be applied to both the master and slave blocks.
INTERFACE_TYPE	String	"NETWORKING", "NETWORKING_PIPELINED", "RETIMED"	"NETWORKING"	Selects mode of operation and determines which set of parallel data is available to the FPGA logic.
SERDES_MODE	String	"NONE", "MASTER", "SLAVE"	"NONE"	Indicates if the ISERDES is being used alone, or as a master or slave, when two ISERDES2 blocks are cascaded.

VHDL Instantiation Template

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

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


-- ISERDES2: Input SERial/DESerializer
--           Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

ISERDES2_inst : ISERDES2
generic map (
   BITSLIP_ENABLE => FALSE,         -- Enable Bitslip Functionality (TRUE/FALSE)
   DATA_RATE => "SDR",              -- Data-rate ("SDR" or "DDR")
   DATA_WIDTH => 1,                 -- Parallel data width selection (2-8)
   INTERFACE_TYPE => "NETWORKING", -- "NETWORKING", "NETWORKING_PIPELINED" or "RETIMED"
   SERDES_MODE => "NONE"            -- "NONE", "MASTER" or "SLAVE"
)
port map (
   CFB0 => CFB0,           -- 1-bit output: Clock feed-through route output
   CFB1 => CFB1,           -- 1-bit output: Clock feed-through route output
   DFB => DFB,             -- 1-bit output: Feed-through clock output
   FABRICOUT => FABRICOUT, -- 1-bit output: Unsynchrnonized data output
   INCDEC => INCDEC,       -- 1-bit output: Phase detector output
   -- Q1 - Q4: 1-bit (each) output: Registered outputs to FPGA logic
   Q1 => Q1,
   Q2 => Q2,
   Q3 => Q3,
   Q4 => Q4,
   SHIFTOUT => SHIFTOUT,   -- 1-bit output: Cascade output signal for master/slave I/O
   VALID => VALID,         -- 1-bit output: Output status of the phase detector
```

```
   BITSLIP => BITSLIP,     -- 1-bit input: Bitslip enable input
   CE0 => CE0,             -- 1-bit input: Clock enable input
   CLK0 => CLK0,           -- 1-bit input: I/O clock network input
   CLK1 => CLK1,           -- 1-bit input: Secondary I/O clock network input
   CLKDIV => CLKDIV,       -- 1-bit input: FPGA logic domain clock input
   D => D,                 -- 1-bit input: Input data
   IOCE => IOCE,           -- 1-bit input: Data strobe input
   RST => RST,             -- 1-bit input: Asynchronous reset input
   SHIFTIN => SHIFTIN      -- 1-bit input: Cascade input signal for master/slave I/O
);

-- End of ISERDES2_inst instantiation
```

Verilog Instantiation Template

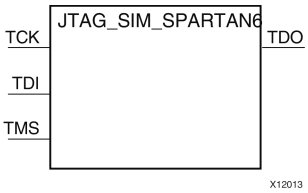
```
// ISERDES2: Input SERial/DESerializer
//           Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

ISERDES2 #(
   .BITSLIP_ENABLE("FALSE"),      // Enable Bitslip Functionality (TRUE/FALSE)
   .DATA_RATE("SDR"),             // Data-rate ("SDR" or "DDR")
   .DATA_WIDTH(1),                // Parallel data width selection (2-8)
   .INTERFACE_TYPE("NETWORKING"), // "NETWORKING", "NETWORKING_PIPELINED" or "RETIMED"
   .SERDES_MODE("NONE")           // "NONE", "MASTER" or "SLAVE"
)
ISERDES2_inst (
   .CFB0(CFB0),           // 1-bit output: Clock feed-through route output
   .CFB1(CFB1),           // 1-bit output: Clock feed-through route output
   .DFB(DFB),             // 1-bit output: Feed-through clock output
   .FABRICOUT(FABRICOUT), // 1-bit output: Unsynchrnonized data output
   .INCDEC(INCDEC),       // 1-bit output: Phase detector output
   // Q1 - Q4: 1-bit (each) output: Registered outputs to FPGA logic
   .Q1(Q1),
   .Q2(Q2),
   .Q3(Q3),
   .Q4(Q4),
   .SHIFTOUT(SHIFTOUT),   // 1-bit output: Cascade output signal for master/slave I/O
   .VALID(VALID),         // 1-bit output: Output status of the phase detector
   .BITSLIP(BITSLIP),     // 1-bit input: Bitslip enable input
   .CE0(CE0),             // 1-bit input: Clock enable input
   .CLK0(CLK0),           // 1-bit input: I/O clock network input
   .CLK1(CLK1),           // 1-bit input: Secondary I/O clock network input
   .CLKDIV(CLKDIV),       // 1-bit input: FPGA logic domain clock input
   .D(D),                 // 1-bit input: Input data
   .IOCE(IOCE),           // 1-bit input: Data strobe input
   .RST(RST),             // 1-bit input: Asynchronous reset input
   .SHIFTIN(SHIFTIN)      // 1-bit input: Cascade input signal for master/slave I/O
);

// End of ISERDES2_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA SelectIO Resources User Guide*
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

JTAG_SIM_SPARTAN6

Simulation: JTAG TAP Controller Simulation Model



Introduction

This simulation component allows the functional simulation of the JTAG TAP controller interface, functions and commands to assist with board-level understanding and debug of the JTAG and Boundary-scan behaviors as well as the behaviors connected to the USER commands and the BSCAN_SPARTAN6 components. This model does not map to a specific primitive in the FPGA software and cannot be directly instantiated in the design however can be used in conjunction with the source design if specified either in a simulation-only file like a testbench or by some means guarded from synthesis so that it is not synthesized into the design netlist. This model may be used for either functional (RTL) simulation or timing simulation. .

Port Descriptions

Port	Direction	Width	Function
TDO	Output	1	**Test Data Out -** This pin is the serial output for all JTAG instruction and data registers. The state of the TAP controller and the current instruction determine the register (instruction or data) that feeds TDO for a specific operation. TDO changes state on the falling edge of TCK and is only active during the shifting of instructions or data through the device. TDO is an active driver output.
TCK	Input	1	**Test Clock -** This pin is the JTAG Test Clock. TCK sequences the TAP controller and the JTAG registers.
TDI	Input	1	**Test Data -** This pin is the serial input to all JTAG instruction and data registers. The state of the TAP controller and the current instruction determine the register that is fed by the TDI pin for a specific operation. TDI has an internal resistive pull-up to provide a logic High to the system if the pin is not driven. TDI is applied into the JTAG registers on the rising edge of TCK.
TMS	Input	1	**Test Mode Select -** This pin determines the sequence of states through the TAP controller on the rising edge of TCK. TMS has an internal resistive pull-up to provide a logic High if the pin is not driven.

Design Entry Method

Instantiation	In testbench or simulation-only file.
Inference	No
CORE Generator™ and wizards	No
Macro support	No

Xilinx suggests that you instantiate this in the testbench file and not an implementation file or file used during synthesis of the design.

More information on simulating and using this component can be found in the *Xilinx Synthesis and Simulation Design Guide*. Please refer to that guide for further detail on using this component.

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
PART_NAME	String	"LX4", "LX9", "LX16", "LX25", "LX25T", "LX45", "LX45T", "LX75", "LX75T", "LX100", "LX100T", "LX150", "LX150T""	"LX4"	Specify the target device in order to properly set IDCODE and other device specific attributes.

VHDL Instantiation Template

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

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


-- JTAG_SIM_SPARTAN6: JTAG Interface Simulation Model
--                    Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

JTAG_SIM_SPARTAN6_inst : JTAG_SIM_SPARTAN6
generic map (
   PART_NAME => "LX4") -- Specify target S6 device. Possible values are:
                       -- "LX4","LX150","LX150T","LX16","LX45","LX45T"
port map (
   TDO => TDO,         -- JTAG data output (1-bit)
   TCK => TCK,         -- Clock input (1-bit)
   TDI => TDI,         -- JTAG data input (1-bit)
   TMS => TMS          -- JTAG command input (1-bit)
);

-- End of JTAG_SIM_SPARTAN6_inst instantiation
```

Verilog Instantiation Template

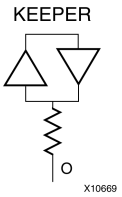
```
// JTAG_SIM_SPARTAN6: JTAG Interface Simulation Model
//                    Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

JTAG_SIM_SPARTAN6 #(
   .PART_NAME("LX4") // Specify target S6 device.  Possible values are:
                     // "LX4","LX150","LX150T","LX16","LX4","LX45","LX45T"
) JTAG_SIM_SPARTAN6_inst (
   .TDO(TDO), // 1-bit JTAG data output
   .TCK(TCK), // 1-bit Clock input
   .TDI(TDI), // 1-bit JTAG data input
   .TMS(TMS)  // 1-bit JTAG command input
);

// End of JTAG_SIM_SPARTAN6_inst instantiation
```

For More Information

- See the *Synthesis and Simulation Design Guide*.
- See the *Spartan-6 FPGA Configuration User Guide*.

KEEPER

Primitive: KEEPER Symbol



Introduction

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

Port Descriptions

Name	Direction	Width	Function
O	Output	1-Bit	Keeper output

Design Entry Method

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

VHDL Instantiation Template

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

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


-- KEEPER: I/O Buffer Weak Keeper
--         Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of KEEPER_inst instantiation
```

Verilog Instantiation Template

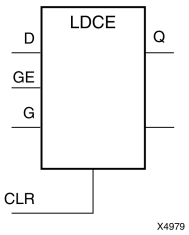
```
// KEEPER: I/O Buffer Weak Keeper
//         Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

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

// End of KEEPER_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA SelectIO Resources User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

LDCE

Primitive: Transparent Data Latch with Asynchronous Clear and Gate Enable



Introduction

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

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

Logic Table

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

Design Entry Method

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

Available Attributes

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

VHDL Instantiation Template

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

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


-- LDCE: Transparent latch with Asynchronous Reset and
--        Gate Enable.
--        Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of LDCE_inst instantiation
```

Verilog Instantiation Template

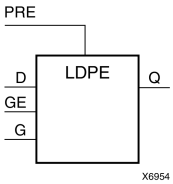
```
// LDCE:  Transparent latch with Asynchronous Reset and Gate Enable.
//        Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

LDCE LDCE_inst (
   .Q(Q),      // Data output
   .CLR(CLR),  // Asynchronous clear/reset input
   .D(D),      // Data input
   .G(G),      // Gate input
   .GE(GE)     // Gate enable input
);

// End of LDCE_inst instantiation
```

For More Information

See the *Spartan-6 FPGA User Documentation (User Guides and Data Sheets)*.

LDPE

Primitive: Transparent Data Latch with Asynchronous Preset and Gate Enable



Introduction

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

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

Logic Table

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

Design Entry Method

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

Available Attributes

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

VHDL Instantiation Template

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

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


-- LDPE:  Transparent latch with Asynchronous Set and
--        Gate Enable.
--        Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of LDPE_inst instantiation
```

Verilog Instantiation Template

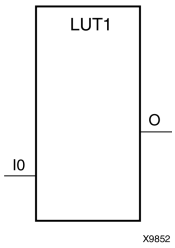
```
// LDPE:  Transparent latch with Asynchronous Preset and Gate Enable.
//        Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

LDPE LDPE_inst (
   .Q(Q),      // Data output
   .PRE(PRE),  // Asynchronous preset/set input
   .D(D),      // Data input
   .G(G),      // Gate input
   .GE(GE)     // Gate enable input
);

// End of LDPE_inst instantiation
```

For More Information

See the *Spartan-6 FPGA User Documentation (User Guides and Data Sheets)*.

LUT1

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



Introduction

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

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

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

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

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

Logic Table

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

Design Entry Method

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

Available Attributes

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

VHDL Instantiation Template

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

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


-- LUT1: 1-input Look-Up Table with general output
--       Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of LUT1_inst instantiation
```

Verilog Instantiation Template

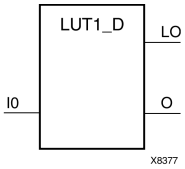
```
// LUT1: 1-input Look-Up Table with general output
//       Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

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

// End of LUT1_inst instantiation
```

For More Information

See the *Spartan-6 FPGA User Documentation (User Guides and Data Sheets)*.

LUT1_D

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



Introduction

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

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

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

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

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

Logic Table

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

Design Entry Method

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

Available Attributes

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

VHDL Instantiation Template

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

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


-- LUT1_D: 1-input Look-Up Table with general and local outputs
--         Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of LUT1_D_inst instantiation
```

Verilog Instantiation Template

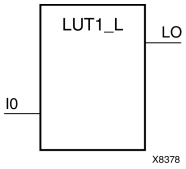
```
// LUT1_D: 1-input Look-Up Table with general and local outputs
//         Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

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

// End of LUT1_D_inst instantiation
```

For More Information

See the *Spartan-6 FPGA User Documentation (User Guides and Data Sheets)*.

LUT1_L

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



Introduction

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

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

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

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

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

Logic Table

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

Design Entry Method

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

Available Attributes

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

VHDL Instantiation Template

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

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


-- LUT1_L: 1-input Look-Up Table with local output
--         Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

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

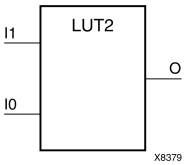
```
// LUT1_L: 1-input Look-Up Table with local output
//         Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

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-6 FPGA User Documentation (User Guides and Data Sheets)*.

LUT2

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



Introduction

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

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

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

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

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

Logic Table

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

Design Entry Method

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

Available Attributes

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

VHDL Instantiation Template

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

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


-- LUT2: 2-input Look-Up Table with general output
--       Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of LUT2_inst instantiation
```

Verilog Instantiation Template

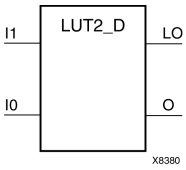
```
// LUT2: 2-input Look-Up Table with general output
//       Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

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

// End of LUT2_inst instantiation
```

For More Information

See the *Spartan-6 FPGA User Documentation (User Guides and Data Sheets)*.

LUT2_D

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



Introduction

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

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

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

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

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

Logic Table

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

Design Entry Method

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

Available Attributes

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

VHDL Instantiation Template

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

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


-- LUT2_D: 2-input Look-Up Table with general and local outputs
--         Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of LUT2_D_inst instantiation
```

Verilog Instantiation Template

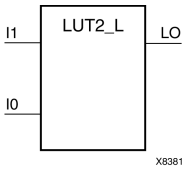
```
// LUT2_D: 2-input Look-Up Table with general and local outputs
//         Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

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

// End of LUT2_L_inst instantiation
```

For More Information

See the *Spartan-6 FPGA User Documentation (User Guides and Data Sheets)*.

LUT2_L

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



Introduction

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

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

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

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

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

Logic Table

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

Design Entry Method

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

Available Attributes

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

VHDL Instantiation Template

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

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


-- LUT2_L: 2-input Look-Up Table with local output
--         Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of LUT2_L_inst instantiation
```

Verilog Instantiation Template

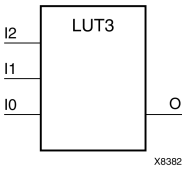
```
// LUT2_L: 2-input Look-Up Table with local output
//         Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

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-6 FPGA User Documentation (User Guides and Data Sheets)*.

LUT3

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



Introduction

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

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

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

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

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

Logic Table

Inputs			Outputs
I2	**I1**	**I0**	**O**
0	0	0	INIT[0]
0	0	1	INIT[1]
0	1	0	INIT[2]
0	1	1	INIT[3]
1	0	0	INIT[4]
1	0	1	INIT[5]
1	1	0	INIT[6]
1	1	1	INIT[7]
INIT = Binary equivalent of the hexadecimal number assigned to the INIT attribute			

Design Entry Method

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

Available Attributes

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

VHDL Instantiation Template

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

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


-- LUT3: 3-input Look-Up Table with general output
--       Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of LUT3_inst instantiation
```

Verilog Instantiation Template

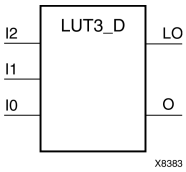
```
// LUT3: 3-input Look-Up Table with general output
//       Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

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

// End of LUT3_inst instantiation
```

For More Information

See the *Spartan-6 FPGA User Documentation (User Guides and Data Sheets)*.

LUT3_D

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



Introduction

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

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

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

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

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

Logic Table

Inputs			Outputs	
I2	**I1**	**I0**	**O**	**LO**
0	0	0	INIT[0]	INIT[0]
0	0	1	INIT[1]	INIT[1]
0	1	0	INIT[2]	INIT[2]
0	1	1	INIT[3]	INIT[3]
1	0	0	INIT[4]	INIT[4]
1	0	1	INIT[5]	INIT[5]
1	1	0	INIT[6]	INIT[6]
1	1	1	INIT[7]	INIT[7]
INIT = Binary equivalent of the hexadecimal number assigned to the INIT attribute				

Design Entry Method

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

Available Attributes

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

VHDL Instantiation Template

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

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


-- LUT3_D: 3-input Look-Up Table with general and local outputs
--         Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of LUT3_D_inst instantiation
```

Verilog Instantiation Template

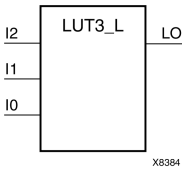
```
// LUT3_D: 3-input Look-Up Table with general and local outputs
//         Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

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

// End of LUT3_D_inst instantiation
```

For More Information

See the *Spartan-6 FPGA User Documentation (User Guides and Data Sheets)*.

LUT3_L

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



Introduction

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

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

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

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

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

Logic Table

Inputs			Outputs
I2	**I1**	**I0**	**LO**
0	0	0	INIT[0]
0	0	1	INIT[1]
0	1	0	INIT[2]
0	1	1	INIT[3]
1	0	0	INIT[4]
1	0	1	INIT[5]
1	1	0	INIT[6]
1	1	1	INIT[7]
INIT = Binary equivalent of the hexadecimal number assigned to the INIT attribute			

Design Entry Method

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

Available Attributes

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

VHDL Instantiation Template

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

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


-- LUT3_L: 3-input Look-Up Table with local output
--         Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of LUT3_L_inst instantiation
```

Verilog Instantiation Template

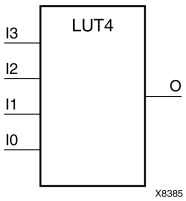
```
// LUT3_L: 3-input Look-Up Table with local output
//         Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

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

// End of LUT3_L_inst instantiation
```

For More Information

See the *Spartan-6 FPGA User Documentation (User Guides and Data Sheets)*.

LUT4

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



Introduction

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

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

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

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

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

Logic Table

Inputs				Outputs
I3	**I2**	**I1**	**I0**	**O**
0	0	0	0	INIT[0]
0	0	0	1	INIT[1]
0	0	1	0	INIT[2]
0	0	1	1	INIT[3]
0	1	0	0	INIT[4]
0	1	0	1	INIT[5]
0	1	1	0	INIT[6]
0	1	1	1	INIT[7]
1	0	0	0	INIT[8]
1	0	0	1	INIT[9]
1	0	1	0	INIT[10]
1	0	1	1	INIT[11]

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

Design Entry Method

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

Available Attributes

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

VHDL Instantiation Template

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

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


-- LUT4: 4-input Look-Up Table with general output
--       Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of LUT4_inst instantiation
```

Verilog Instantiation Template

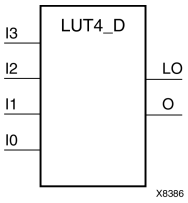
```
// LUT4: 4-input Look-Up Table with general output
//       Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

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-6 FPGA User Documentation (User Guides and Data Sheets)*.

LUT4_D

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



Introduction

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

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

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

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

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

Logic Table

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

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

Design Entry Method

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

Available Attributes

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

VHDL Instantiation Template

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

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


-- LUT4_D: 4-input Look-Up Table with general and local outputs
--         Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of LUT4_D_inst instantiation
```

Verilog Instantiation Template

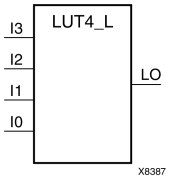
```
// LUT4_D: 4-input Look-Up Table with general and local outputs
//         Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

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-6 FPGA User Documentation (User Guides and Data Sheets)*.

LUT4_L

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



Introduction

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

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

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

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

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

Logic Table

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

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

Design Entry Method

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

Available Attributes

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

VHDL Instantiation Template

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

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


-- LUT4_L: 4-input Look-Up Table with local output
--         Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of LUT4_L_inst instantiation
```

Verilog Instantiation Template

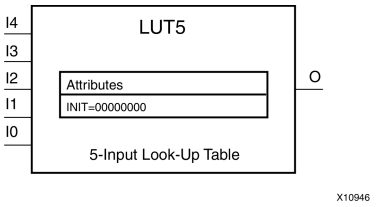
```
// LUT4_L: 4-input Look-Up Table with local output
//         Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

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

// End of LUT4_L_inst instantiation
```

For More Information

See the *Spartan-6 FPGA User Documentation (User Guides and Data Sheets)*.

LUT5

Primitive: 5-Input Lookup Table with General Output



Introduction

This design element is a 5-input, 1-output look-up table (LUT) that can either act as an asynchronous 32-bit ROM (with 5-bit addressing) or implement any 5-input logic function. LUTs are the basic logic building blocks and are used to implement most logic functions of the design. One LUT5 is packed into a LUT6 within a slice, or two LUT5s can be packed into a single LUT6 with some restrictions. The functionality of the LUT5, LUT5_L and LUT5_D is the same. However, the LUT5_L and LUT5_D allow the additional specification to connect the LUT5 output signal to an internal slice or CLB connection using the LO output. The LUT5_L specifies that the only connections from the LUT5 will be within a slice or CLB, while the LUT5_D allows the specification to connect the output of the LUT to both inter-slice/CLB logic and external logic as well. The LUT5 does not state any specific output connections and should be used in all cases except where internal slice or CLB signal connections must be implicitly specified.

An INIT attribute consisting of a 32-bit hexadecimal value must be specified to indicate the LUTs logical function. The INIT value is calculated by assigning a 1 to the corresponding INIT bit value when the associated inputs are applied. For instance, a Verilog INIT value of 32'h80000000 (X"80000000" for VHDL) makes the output zero unless all of the inputs are one (a 5-input AND gate). A Verilog INIT value of 32'hfffffffe (X"FFFFFFFE" for VHDL) makes the output one unless all zeros are on the inputs (a 5-input OR gate).

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

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

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

Logic Table

Inputs					Outputs
I4	I3	I2	I1	I0	LO
0	0	0	0	0	INIT[0]
0	0	0	0	1	INIT[1]
0	0	0	1	0	INIT[2]
0	0	0	1	1	INIT[3]
0	0	1	0	0	INIT[4]
0	0	1	0	1	INIT[5]

Inputs					Outputs
I4	**I3**	**I2**	**I1**	**I0**	**LO**
0	0	1	1	0	INIT[6]
0	0	1	1	1	INIT[7]
0	1	0	0	0	INIT[8]
0	1	0	0	1	INIT[9]
0	1	0	1	0	INIT[10]
0	1	0	1	1	INIT[11]
0	1	1	0	0	INIT[12]
0	1	1	0	1	INIT[13]
0	1	1	1	0	INIT[14]
0	1	1	1	1	INIT[15]
1	0	0	0	0	INIT[16]
1	0	0	0	1	INIT[17]
1	0	0	1	0	INIT[18]
1	0	0	1	1	INIT[19]
1	0	1	0	0	INIT[20]
1	0	1	0	1	INIT[21]
1	0	1	1	0	INIT[22]
1	0	1	1	1	INIT[23]
1	1	0	0	0	INIT[24]
1	1	0	0	1	INIT[25]
1	1	0	1	0	INIT[26]
1	1	0	1	1	INIT[27]
1	1	1	0	0	INIT[28]
1	1	1	0	1	INIT[29]
1	1	1	1	0	INIT[30]
1	1	1	1	1	INIT[31]
INIT = Binary equivalent of the hexadecimal number assigned to the INIT attribute					

Port Description

Name	Direction	Width	Function
O	Output	1	5-LUT output
I0, I1, I2, I3, I4	Input	1	LUT inputs

Design Entry Method

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

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
INIT	Hexadecimal	Any 32-Bit Value	All zeros	Specifies the logic value for the 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;


-- LUT5: 5-input Look-Up Table with general output
--       Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

LUT5_inst : LUT5
generic map (
   INIT => X"00000000") -- Specify LUT Contents
port map (
   O => O,  -- LUT general output
   I0 => I0,   -- LUT input
   I1 => I1,   -- LUT input
   I2 => I2,   -- LUT input
   I3 => I3,   -- LUT input
   I4 => I4    -- LUT input
);

-- End of LUT5_inst instantiation
```

Verilog Instantiation Template

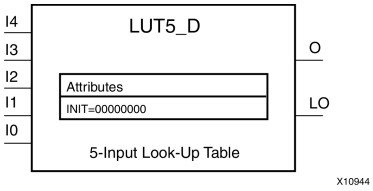
```
// LUT5: 5-input Look-Up Table with general output
//       Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

LUT5 #(
   .INIT(32'h00000000)  // Specify LUT Contents
) LUT5_inst (
   .O(O),   // LUT general output
   .I0(I0), // LUT input
   .I1(I1), // LUT input
   .I2(I2), // LUT input
   .I3(I3), // LUT input
   .I4(I4)  // LUT input
);

// End of LUT5_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA Configurable Logic Block User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

LUT5_D

Primitive: 5-Input Lookup Table with General and Local Outputs



Introduction

This design element is a 5-input, 1-output look-up table (LUT) that can either act as an asynchronous 32-bit ROM (with 5-bit addressing) or implement any 5-input logic function. LUTs are the basic logic building blocks and are used to implement most logic functions of the design. One LUT5 will be packed into a LUT6 within a slice, or two LUT5s can be packed into a single LUT6 with some restrictions. The functionality of the LUT5, LUT5_L and LUT5_D is the same. However, the LUT5_L and LUT5_D allow the additional specification to connect the LUT5 output signal to an internal slice or CLB connection using the LO output. The LUT5_L specifies that the only connections from the LUT5 will be within a slice or CLB, while the LUT5_D allows the specification to connect the output of the LUT to both inter-slice/CLB logic and external logic as well. The LUT5 does not state any specific output connections and should be used in all cases except where internal slice or CLB signal connections must be implicitly specified.

An INIT attribute consisting of a 32-bit hexadecimal value must be specified to indicate the LUTs logical function. The INIT value is calculated by assigning a 1 to the corresponding INIT bit value when the associated inputs are applied. For instance, a Verilog INIT value of 32'h80000000 (X"80000000" for VHDL) will make the output zero unless all of the inputs are one (a 5-input AND gate). A Verilog INIT value of 32'hfffffffe (X"FFFFFFFE" for VHDL) will make the output one unless all zeros are on the inputs (a 5-input OR gate).

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

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

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

Logic Table

Inputs					Outputs	
I4	I3	I2	I1	I0	O	LO
0	0	0	0	0	INIT[0]	INIT[0]
0	0	0	0	1	INIT[1]	INIT[1]
0	0	0	1	0	INIT[2]	INIT[2]
0	0	0	1	1	INIT[3]	INIT[3]
0	0	1	0	0	INIT[4]	INIT[4]
0	0	1	0	1	INIT[5]	INIT[5]

Inputs					Outputs	
I4	I3	I2	I1	I0	O	LO
0	0	1	1	0	INIT[6]	INIT[6]
0	0	1	1	1	INIT[7]	INIT[7]
0	1	0	0	0	INIT[8]	INIT[8]
0	1	0	0	1	INIT[9]	INIT[9]
0	1	0	1	0	INIT[10]	INIT[10]
0	1	0	1	1	INIT[11]	INIT[11]
0	1	1	0	0	INIT[12]	INIT[12]
0	1	1	0	1	INIT[13]	INIT[13]
0	1	1	1	0	INIT[14]	INIT[14]
0	1	1	1	1	INIT[15]	INIT[15]
1	0	0	0	0	INIT[16]	INIT[16]
1	0	0	0	1	INIT[17]	INIT[17]
1	0	0	1	0	INIT[18]	INIT[18]
1	0	0	1	1	INIT[19]	INIT[19]
1	0	1	0	0	INIT[20]	INIT[20]
1	0	1	0	1	INIT[21]	INIT[21]
1	0	1	1	0	INIT[22]	INIT[22]
1	0	1	1	1	INIT[23]	INIT[23]
1	1	0	0	0	INIT[24]	INIT[24]
1	1	0	0	1	INIT[25]	INIT[25]
1	1	0	1	0	INIT[26]	INIT[26]
1	1	0	1	1	INIT[27]	INIT[27]
1	1	1	0	0	INIT[28]	INIT[28]
1	1	1	0	1	INIT[29]	INIT[29]
1	1	1	1	0	INIT[30]	INIT[30]
1	1	1	1	1	INIT[31]	INIT[31]
INIT = Binary equivalent of the hexadecimal number assigned to the INIT attribute						

Port Description

Name	Direction	Width	Function
O	Output	1	5-LUT output
L0	Output	1	5-LUT output for internal CLB connection
I0, I1, I2, I3, I4	Input	1	LUT inputs

Design Entry Method

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

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
INIT	Hexadecimal	Any 32-Bit Value	All zeros	Specifies the logic value for the 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;


-- LUT5_D: 5-input Look-Up Table with general and local outputs
--         Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

LUT5_D_inst : LUT5_D
generic map (
   INIT => X"00000000") -- Specify LUT contents
port map (
   LO => LO, -- LUT local output
   O => O,   -- LUT general output
   I0 => I0, -- LUT input
   I1 => I1, -- LUT input
   I2 => I2, -- LUT input
   I3 => I3, -- LUT input
   I4 => I4  -- LUT input
);

-- End of LUT5_D_inst instantiation
```

Verilog Instantiation Template

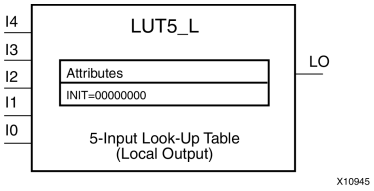
```
// LUT5_D: 5-input Look-Up Table with general and local outputs
//         Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

LUT5_D #(
   .INIT(32'h0000000)  // Specify LUT Contents
) LUT5_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
   .I4(I4)  // LUT input
);

// End of LUT5_D_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA Configurable Logic Block User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

LUT5_L

Primitive: 5-Input Lookup Table with Local Output



Introduction

This design element is a 5-input, 1-output look-up table (LUT) that can either act as an asynchronous 32-bit ROM (with 5-bit addressing) or implement any 5-input logic function. LUTs are the basic logic building blocks and are used to implement most logic functions of the design. One LUT5 will be packed into a LUT6 within a slice, or two LUT5s can be packed into a single LUT6 with some restrictions. The functionality of the LUT5, LUT5_L and LUT5_D is the same. However, the LUT5_L and LUT5_D allow the additional specification to connect the LUT5 output signal to an internal slice or CLB connection using the LO output. The LUT5_L specifies that the only connections from the LUT5 is within a slice or CLB, while the LUT5_D allows the specification to connect the output of the LUT to both inter-slice/CLB logic and external logic as well. The LUT5 does not state any specific output connections and should be used in all cases except where internal slice or CLB signal connections must be implicitly specified.

An INIT attribute consisting of a 32-bit hexadecimal value must be specified to indicate the LUTs logical function. The INIT value is calculated by assigning a 1 to the corresponding INIT bit value when the associated inputs are applied. For instance, a Verilog INIT value of 32'h80000000 (X"80000000" for VHDL) makes the output zero unless all of the inputs are one (a 5-input AND gate). A Verilog INIT value of 32'hfffffffe (X"FFFFFFFE" for VHDL) makes the output one unless all zeros are on the inputs (a 5-input OR gate).

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

The Logic Table Method -A common method to determine the desired INIT value for a LUT is using a logic table. To do so, simply create a binary 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 logic value and use those to build the logic equation you are after. This method is easier to understand once you have grasped the concept and is more self-documenting than the above method. However, this method does require the code to first specify the appropriate parameters.

Logic Table

Inputs					Outputs
I4	I3	I2	I1	I0	LO
0	0	0	0	0	INIT[0]
0	0	0	0	1	INIT[1]
0	0	0	1	0	INIT[2]
0	0	0	1	1	INIT[3]
0	0	1	0	0	INIT[4]
0	0	1	0	1	INIT[5]

Inputs					Outputs
I4	**I3**	**I2**	**I1**	**I0**	**LO**
0	0	1	1	0	INIT[6]
0	0	1	1	1	INIT[7]
0	1	0	0	0	INIT[8]
0	1	0	0	1	INIT[9]
0	1	0	1	0	INIT[10]
0	1	0	1	1	INIT[11]
0	1	1	0	0	INIT[12]
0	1	1	0	1	INIT[13]
0	1	1	1	0	INIT[14]
0	1	1	1	1	INIT[15]
1	0	0	0	0	INIT[16]
1	0	0	0	1	INIT[17]
1	0	0	1	0	INIT[18]
1	0	0	1	1	INIT[19]
1	0	1	0	0	INIT[20]
1	0	1	0	1	INIT[21]
1	0	1	1	0	INIT[22]
1	0	1	1	1	INIT[23]
1	1	0	0	0	INIT[24]
1	1	0	0	1	INIT[25]
1	1	0	1	0	INIT[26]
1	1	0	1	1	INIT[27]
1	1	1	0	0	INIT[28]
1	1	1	0	1	INIT[29]
1	1	1	1	0	INIT[30]
1	1	1	1	1	INIT[31]
INIT = Binary equivalent of the hexadecimal number assigned to the INIT attribute					

Port Description

Name	Direction	Width	Function
L0	Output	1	6/5-LUT output for internal CLB connection
I0, I1, I2, I3, I4	Input	1	LUT inputs

Design Entry Method

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

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
INIT	Hexadecimal	Any 32-Bit Value	All zeros	Specifies the logic value for the 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;


-- LUT5_L: 5-input Look-Up Table with local output
--         Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

LUT5_L_inst : LUT5_L
generic map (
   INIT => X"00000000") -- Specify LUT Contents
port map (
   LO => LO, -- LUT local output
   I0 => I0, -- LUT input
   I1 => I1, -- LUT input
   I2 => I2, -- LUT input
   I3 => I3, -- LUT input
   I4 => I4  -- LUT input
);

-- End of LUT5_L_inst instantiation
```

Verilog Instantiation Template

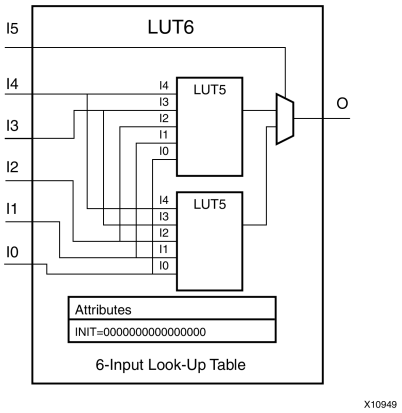
```
// LUT5_L: 5-input Look-Up Table with local output
//         Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

LUT5_L #(
   .INIT(32'h0000000)  // Specify LUT Contents
) LUT5_L_inst (
   .LO(LO), // LUT local output
   .I0(I0), // LUT input
   .I1(I1), // LUT input
   .I2(I2), // LUT input
   .I3(I3), // LUT input
   .I4(I4)  // LUT input
);

// End of LUT5_L_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA Configurable Logic Block User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

LUT6

Primitive: 6-Input Lookup Table with General Output



Introduction

This design element is a 6-input, 1-output look-up table (LUT) that can either act as an asynchronous 64-bit ROM (with 6-bit addressing) or implement any 6-input logic function. LUTs are the basic logic building blocks and are used to implement most logic functions of the design. A LUT6 is mapped to one of the four look-up tables in the slice. The functionality of the LUT6, LUT6_L and LUT6_D is the same. However, the LUT6_L and LUT6_D allow the additional specification to connect the LUT6 output signal to an internal slice, or CLB connection, using the LO output. The LUT6_L specifies that the only connections from the LUT6 will be within a slice, or CLB, while the LUT6_D allows the specification to connect the output of the LUT to both inter-slice/CLB logic and external logic as well. The LUT6 does not state any specific output connections and should be used in all cases except where internal slice or CLB signal connections must be implicitly specified.

An INIT attribute consisting of a 64-bit Hexadecimal value must be specified to indicate the LUTs logical function. The INIT value is calculated by assigning a 1 to corresponding INIT bit value when the associated inputs are applied. For instance, a Verilog INIT value of 64'h8000000000000000 (X"8000000000000000" for VHDL) makes the output zero unless all of the inputs are one (a 6-input AND gate). A Verilog INIT value of 64'hfffffffffffffffe (X"FFFFFFFFFFFFFFFE" for VHDL) makes the output one unless all zeros are on the inputs (a 6-input OR gate).

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

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

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

Logic Table

Inputs						Outputs
I5	I4	I3	I2	I1	I0	O
0	0	0	0	0	0	INIT[0]
0	0	0	0	0	1	INIT[1]
0	0	0	0	1	0	INIT[2]

Inputs						Outputs
I5	**I4**	**I3**	**I2**	**I1**	**I0**	**O**
0	0	0	0	1	1	INIT[3]
0	0	0	1	0	0	INIT[4]
0	0	0	1	0	1	INIT[5]
0	0	0	1	1	0	INIT[6]
0	0	0	1	1	1	INIT[7]
0	0	1	0	0	0	INIT[8]
0	0	1	0	0	1	INIT[9]
0	0	1	0	1	0	INIT[10]
0	0	1	0	1	1	INIT[11]
0	0	1	1	0	0	INIT[12]
0	0	1	1	0	1	INIT[13]
0	0	1	1	1	0	INIT[14]
0	0	1	1	1	1	INIT[15]
0	1	0	0	0	0	INIT[16]
0	1	0	0	0	1	INIT[17]
0	1	0	0	1	0	INIT[18]
0	1	0	0	1	1	INIT[19]
0	1	0	1	0	0	INIT[20]
0	1	0	1	0	1	INIT[21]
0	1	0	1	1	0	INIT[22]
0	1	0	1	1	1	INIT[23]
0	1	1	0	0	0	INIT[24]
0	1	1	0	0	1	INIT[25]
0	1	1	0	1	0	INIT[26]
0	1	1	0	1	1	INIT[27]
0	1	1	1	0	0	INIT[28]
0	1	1	1	0	1	INIT[29]
0	1	1	1	1	0	INIT[30]
0	1	1	1	1	1	INIT[31]
1	0	0	0	0	0	INIT[32]
1	0	0	0	0	1	INIT[33]
1	0	0	0	1	0	INIT[34]
1	0	0	0	1	1	INIT[35]
1	0	0	1	0	0	INIT[36]
1	0	0	1	0	1	INIT[37]
1	0	0	1	1	0	INIT[38]
1	0	0	1	1	1	INIT[39]

Inputs						Outputs
I5	I4	I3	I2	I1	I0	O
1	0	1	0	0	0	INIT[40]
1	0	1	0	0	1	INIT[41]
1	0	1	0	1	0	INIT[42]
1	0	1	0	1	1	INIT[43]
1	0	1	1	0	0	INIT[44]
1	0	1	1	0	1	INIT[45]
1	0	1	1	1	0	INIT[46]
1	0	1	1	1	1	INIT[47]
1	1	0	0	0	0	INIT[48]
1	1	0	0	0	1	INIT[49]
1	1	0	0	1	0	INIT[50]
1	1	0	0	1	1	INIT[51]
1	1	0	1	0	0	INIT[52]
1	1	0	1	0	1	INIT[53]
1	1	0	1	1	0	INIT[54]
1	1	0	1	1	1	INIT[55]
1	1	1	0	0	0	INIT[56]
1	1	1	0	0	1	INIT[57]
1	1	1	0	1	0	INIT[58]
1	1	1	0	1	1	INIT[59]
1	1	1	1	0	0	INIT[60]
1	1	1	1	0	1	INIT[61]
1	1	1	1	1	0	INIT[62]
1	1	1	1	1	1	INIT[63]
INIT = Binary equivalent of the hexadecimal number assigned to the INIT attribute						

Port Description

Name	Direction	Width	Function
O	Output	1	6/5-LUT output
I0, I1, I2, I3, I4, I5	Input	1	LUT inputs

Design Entry Method

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

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
INIT	Hexadecimal	Any 64-Bit Value	All zeros	Specifies the logic value for the 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;


-- LUT6: 6-input Look-Up Table with general output
--       Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

LUT6_inst : LUT6
generic map (
   INIT => X"0000000000000000") -- Specify LUT Contents
port map (
   O => O,  -- LUT general output
   I0 => I0,   -- LUT input
   I1 => I1,   -- LUT input
   I2 => I2,   -- LUT input
   I3 => I3,   -- LUT input
   I4 => I4,   -- LUT input
   I5 => I5    -- LUT input
);

-- End of LUT6_inst instantiation
```

Verilog Instantiation Template

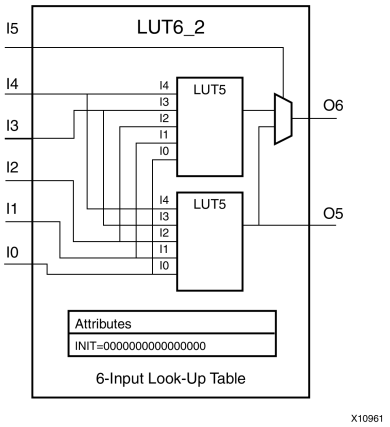
```
// LUT6: 6-input Look-Up Table with general output
//       Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

LUT6 #(
   .INIT(64'h0000000000000000)  // Specify LUT Contents
) LUT6_inst (
   .O(O),   // LUT general output
   .I0(I0), // LUT input
   .I1(I1), // LUT input
   .I2(I2), // LUT input
   .I3(I3), // LUT input
   .I4(I4), // LUT input
   .I5(I5)  // LUT input
);

// End of LUT6_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA Configurable Logic Block User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

LUT6_2

Primitive: Six-input, 2-output, Look-Up Table



Introduction

This design element is a 6-input, 2-output look-up table (LUT) that can either act as a dual asynchronous 32-bit ROM (with 5-bit addressing), implement any two 5-input logic functions with shared inputs, or implement a 6-input logic function and a 5-input logic function with shared inputs and shared logic values. LUTs are the basic logic building blocks and are used to implement most logic functions of the design. A LUT6_2 will be mapped to one of the four look-up tables in the slice.

An INIT attribute consisting of a 64-bit hexadecimal value must be specified to indicate the LUTs logical function. The INIT value is calculated by assigning a 1 to corresponding INIT bit value when the associated inputs are applied. For instance, a Verilog INIT value of 64'hfffffffffffffffe (X"FFFFFFFFFFFFFFFE" for VHDL) makes the O6 output 1 unless all zeros are on the inputs and the O5 output a 1, or unless I[4:0] are all zeroes (a 5-input and 6-input OR gate). The lower half (bits 31:0) of the INIT values apply to the logic function of the O5 output.

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

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

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

Logic Table

Inputs						Outputs	
I5	I4	I3	I2	I1	I0	O5	O6
0	0	0	0	0	0	INIT[0]	INIT[0]
0	0	0	0	0	1	INIT[1]	INIT[1]
0	0	0	0	1	0	INIT[2]	INIT[2]
0	0	0	0	1	1	INIT[3]	INIT[3]
0	0	0	1	0	0	INIT[4]	INIT[4]

Inputs						Outputs	
0	0	0	1	0	1	INIT[5]	INIT[5]
0	0	0	1	1	0	INIT[6]	INIT[6]
0	0	0	1	1	1	INIT[7]	INIT[7]
0	0	1	0	0	0	INIT[8]	INIT[8]
0	0	1	0	0	1	INIT[9]	INIT[9]
0	0	1	0	1	0	INIT[10]	INIT[10]
0	0	1	0	1	1	INIT[11]	INIT[11]
0	0	1	1	0	0	INIT[12]	INIT[12]
0	0	1	1	0	1	INIT[13]	INIT[13]
0	0	1	1	1	0	INIT[14]	INIT[14]
0	0	1	1	1	1	INIT[15]	INIT[15]
0	1	0	0	0	0	INIT[16]	INIT[16]
0	1	0	0	0	1	INIT[17]	INIT[17]
0	1	0	0	1	0	INIT[18]	INIT[18]
0	1	0	0	1	1	INIT[19]	INIT[19]
0	1	0	1	0	0	INIT[20]	INIT[20]
0	1	0	1	0	1	INIT[21]	INIT[21]
0	1	0	1	1	0	INIT[22]	INIT[22]
0	1	0	1	1	1	INIT[23]	INIT[23]
0	1	1	0	0	0	INIT[24]	INIT[24]
0	1	1	0	0	1	INIT[25]	INIT[25]
0	1	1	0	1	0	INIT[26]	INIT[26]
0	1	1	0	1	1	INIT[27]	INIT[27]
0	1	1	1	0	0	INIT[28]	INIT[28]
0	1	1	1	0	1	INIT[29]	INIT[29]
0	1	1	1	1	0	INIT[30]	INIT[30]
0	1	1	1	1	1	INIT[31]	INIT[31]
1	0	0	0	0	0	INIT[0]	INIT[32]
1	0	0	0	0	1	INIT[1]	INIT[33]
1	0	0	0	1	0	INIT[2]	INIT[34]
1	0	0	0	1	1	INIT[3]	INIT[35]
1	0	0	1	0	0	INIT[4]	INIT[36]
1	0	0	1	0	1	INIT[5]	INIT[37]
1	0	0	1	1	0	INIT[6]	INIT[38]
1	0	0	1	1	1	INIT[7]	INIT[39]
1	0	1	0	0	0	INIT[8]	INIT[40]
1	0	1	0	0	1	INIT[9]	INIT[41]
1	0	1	0	1	0	INIT[10]	INIT[42]
1	0	1	0	1	1	INIT[11]	INIT[43]

Inputs						Outputs	
1	0	1	1	0	0	INIT[12]	INIT[44]
1	0	1	1	0	1	INIT[13]	INIT[45]
1	0	1	1	1	0	INIT[14]	INIT[46]
1	0	1	1	1	1	INIT[15]	INIT[47]
1	1	0	0	0	0	INIT[16]	INIT[48]
1	1	0	0	0	1	INIT[17]	INIT[49]
1	1	0	0	1	0	INIT[18]	INIT[50]
1	1	0	0	1	1	INIT[19]	INIT[51]
1	1	0	1	0	0	INIT[20]	INIT[52]
1	1	0	1	0	1	INIT[21]	INIT[53]
1	1	0	1	1	0	INIT[22]	INIT[54]
1	1	0	1	1	1	INIT[23]	INIT[55]
1	1	1	0	0	0	INIT[24]	INIT[56]
1	1	1	0	0	1	INIT[25]	INIT[57]
1	1	1	0	1	0	INIT[26]	INIT[58]
1	1	1	0	1	1	INIT[27]	INIT[59]
1	1	1	1	0	0	INIT[28]	INIT[60]
1	1	1	1	0	1	INIT[29]	INIT[61]
1	1	1	1	1	0	INIT[30]	INIT[62]
1	1	1	1	1	1	INIT[31]	INIT[63]
INIT = Binary equivalent of the hexadecimal number assigned to the INIT attribute							

Port Descriptions

Port	Direction	Width	Function
O6	Output	1	6/5-LUT output
O5	Output	1	5-LUT output
I0, I1, I2, I3, I4, I5	Input	1	LUT inputs

Design Entry Method

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

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
INIT	Hexadecimal	Any 64-Bit Value	All zeros	Specifies the LUT5/6 output function.

VHDL Instantiation Template

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

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


-- LUT6_2: 6-input  2 output Look-Up Table
--         Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of LUT6_2_inst instantiation
```

Verilog Instantiation Template

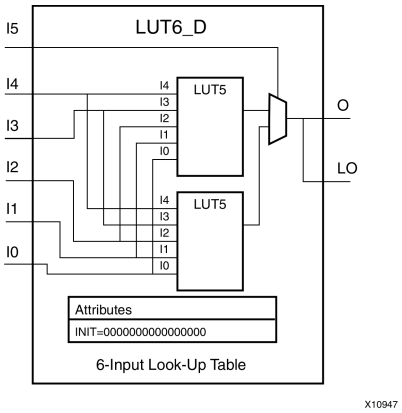
```
// LUT6_2: 6-input, 2 output Look-Up Table
//         Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

LUT6_2 #(
   .INIT(64'h0000000000000000) // Specify LUT Contents
) LUT6_2_inst (
   .O6(O6), // 1-bit LUT6 output
   .O5(O5), // 1-bit lower LUT5 output
   .I0(I0), // 1-bit LUT input
   .I1(I1), // 1-bit LUT input
   .I2(I2), // 1-bit LUT input
   .I3(I3), // 1-bit LUT input
   .I4(I4), // 1-bit LUT input
   .I5(I5)  // 1-bit LUT input (fast MUX select only available to O6 output)
);

// End of LUT6_2_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA Configurable Logic Block User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

LUT6_D

Primitive: 6-Input Lookup Table with General and Local Outputs



Introduction

This design element is a six-input, one-output look-up table (LUT) that can either act as an asynchronous 64-bit ROM (with 6-bit addressing) or implement any 6-input logic function. LUTs are the basic logic building blocks and are used to implement most logic functions of the design. A LUT6 is mapped to one of the four look-up tables in the slice. The functionality of the LUT6, LUT6_L and LUT6_D is the same. However, the LUT6_L and LUT6_D allow the additional specification to connect the LUT6 output signal to an internal slice, or CLB connection, using the LO output. The LUT6_L specifies that the only connections from the LUT6 will be within a slice, or CLB, while the LUT6_D allows the specification to connect the output of the LUT to both inter-slice/CLB logic and external logic as well. The LUT6 does not state any specific output connections and should be used in all cases except where internal slice or CLB signal connections must be implicitly specified.

An INIT attribute consisting of a 64-bit Hexadecimal value must be specified to indicate the LUTs logical function. The INIT value is calculated by assigning a 1 to corresponding INIT bit value when the associated inputs are applied. For instance, a Verilog INIT value of 64'h8000000000000000 (X"8000000000000000" for VHDL) makes the output zero unless all of the inputs are one (a 6-input AND gate). A Verilog INIT value of 64'hfffffffffffffffe (X"FFFFFFFFFFFFFFFE" for VHDL) makes the output one unless all zeros are on the inputs (a 6-input OR gate).

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

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

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

Logic Table

Inputs						Outputs	
I5	**I4**	**I3**	**I2**	**I1**	**I0**	**O**	**LO**
0	0	0	0	0	0	INIT[0]	INIT[0]
0	0	0	0	0	1	INIT[1]	INIT[1]
0	0	0	0	1	0	INIT[2]	INIT[2]

Inputs						Outputs	
I5	**I4**	**I3**	**I2**	**I1**	**I0**	**O**	**LO**
0	0	0	0	1	1	INIT[3]	INIT[3]
0	0	0	1	0	0	INIT[4]	INIT[4]
0	0	0	1	0	1	INIT[5]	INIT[5]
0	0	0	1	1	0	INIT[6]	INIT[6]
0	0	0	1	1	1	INIT[7]	INIT[7]
0	0	1	0	0	0	INIT[8]	INIT[8]
0	0	1	0	0	1	INIT[9]	INIT[9]
0	0	1	0	1	0	INIT[10]	INIT[10]
0	0	1	0	1	1	INIT[11]	INIT[11]
0	0	1	1	0	0	INIT[12]	INIT[12]
0	0	1	1	0	1	INIT[13]	INIT[13]
0	0	1	1	1	0	INIT[14]	INIT[14]
0	0	1	1	1	1	INIT[15]	INIT[15]
0	1	0	0	0	0	INIT[16]	INIT[16]
0	1	0	0	0	1	INIT[17]	INIT[17]
0	1	0	0	1	0	INIT[18]	INIT[18]
0	1	0	0	1	1	INIT[19]	INIT[19]
0	1	0	1	0	0	INIT[20]	INIT[20]
0	1	0	1	0	1	INIT[21]	INIT[21]
0	1	0	1	1	0	INIT[22]	INIT[22]
0	1	0	1	1	1	INIT[23]	INIT[23]
0	1	1	0	0	0	INIT[24]	INIT[24]
0	1	1	0	0	1	INIT[25]	INIT[25]
0	1	1	0	1	0	INIT[26]	INIT[26]
0	1	1	0	1	1	INIT[27]	INIT[27]
0	1	1	1	0	0	INIT[28]	INIT[28]
0	1	1	1	0	1	INIT[29]	INIT[29]
0	1	1	1	1	0	INIT[30]	INIT[30]
0	1	1	1	1	1	INIT[31]	INIT[31]
1	0	0	0	0	0	INIT[32]	INIT[32]
1	0	0	0	0	1	INIT[33]	INIT[33]
1	0	0	0	1	0	INIT[34]	INIT[34]
1	0	0	0	1	1	INIT[35]	INIT[35]
1	0	0	1	0	0	INIT[36]	INIT[36]
1	0	0	1	0	1	INIT[37]	INIT[37]
1	0	0	1	1	0	INIT[38]	INIT[38]
1	0	0	1	1	1	INIT[39]	INIT[39]

Inputs						Outputs	
I5	**I4**	**I3**	**I2**	**I1**	**I0**	**O**	**LO**
1	0	1	0	0	0	INIT[40]	INIT[40]
1	0	1	0	0	1	INIT[41]	INIT[41]
1	0	1	0	1	0	INIT[42]	INIT[42]
1	0	1	0	1	1	INIT[43]	INIT[43]
1	0	1	1	0	0	INIT[44]	INIT[44]
1	0	1	1	0	1	INIT[45]	INIT[45]
1	0	1	1	1	0	INIT[46]	INIT[46]
1	0	1	1	1	1	INIT[47]	INIT[47]
1	1	0	0	0	0	INIT[48]	INIT[48]
1	1	0	0	0	1	INIT[49]	INIT[49]
1	1	0	0	1	0	INIT[50]	INIT[50]
1	1	0	0	1	1	INIT[51]	INIT[51]
1	1	0	1	0	0	INIT[52]	INIT[52]
1	1	0	1	0	1	INIT[53]	INIT[53]
1	1	0	1	1	0	INIT[54]	INIT[54]
1	1	0	1	1	1	INIT[55]	INIT[55]
1	1	1	0	0	0	INIT[56]	INIT[56]
1	1	1	0	0	1	INIT[57]	INIT[57]
1	1	1	0	1	0	INIT[58]	INIT[58]
1	1	1	0	1	1	INIT[59]	INIT[59]
1	1	1	1	0	0	INIT[60]	INIT[60]
1	1	1	1	0	1	INIT[61]	INIT[61]
1	1	1	1	1	0	INIT[62]	INIT[62]
1	1	1	1	1	1	INIT[63]	INIT[63]
INIT = Binary equivalent of the hexadecimal number assigned to the INIT attribute							

Port Description

Name	Direction	Width	Function
O6	Output	1	6/5-LUT output
O5	Output	1	5-LUT output
I0, I1, I2, I3, I4, I5	Input	1	LUT inputs

Design Entry Method

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

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
INIT	Hexadecimal	Any 64-Bit Value	All zeros	Specifies the logic value for the 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;


-- LUT6_D: 6-input Look-Up Table with general and local outputs
--         Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

LUT6_D_inst : LUT6_D
generic map (
   INIT => X"0000000000000000") -- Specify LUT contents
port map (
   LO => LO, -- LUT local output
   O => O,   -- LUT general output
   I0 => I0, -- LUT input
   I1 => I1, -- LUT input
   I2 => I2, -- LUT input
   I3 => I3, -- LUT input
   I4 => I4, -- LUT input
   I5 => I5 -- LUT input
);

-- End of LUT6_D_inst instantiation
```

Verilog Instantiation Template

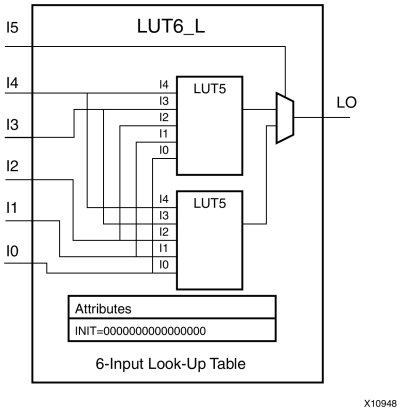
```
// LUT6_D: 6-input Look-Up Table with general and local outputs
//         Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

LUT6_D #(
   .INIT(64'h0000000000000000)  // Specify LUT Contents
) LUT6_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
   .I4(I4), // LUT input
   .I5(I5)  // LUT input
);

// End of LUT6_D_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA Configurable Logic Block User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

LUT6_L

Primitive: 6-Input Lookup Table with Local Output



Introduction

This design element is a 6-input, 1-output look-up table (LUT) that can either act as an asynchronous 64-bit ROM (with 6-bit addressing) or implement any 6-input logic function. LUTs are the basic logic building blocks and are used to implement most logic functions of the design. A LUT6 is mapped to one of the four look-up tables in the slice. The functionality of the LUT6, LUT6_L and LUT6_D is the same. However, the LUT6_L and LUT6_D allow the additional specification to connect the LUT6 output signal to an internal slice, or CLB connection, using the LO output. The LUT6_L specifies that the only connections from the LUT6 are within a slice, or CLB, while the LUT6_D allows the specification to connect the output of the LUT to both inter-slice/CLB logic and external logic as well. The LUT6 does not state any specific output connections and should be used in all cases except where internal slice or CLB signal connections must be implicitly specified.

An INIT attribute consisting of a 64-bit hexadecimal value must be specified to indicate the LUT's logical function. The INIT value is calculated by assigning a 1 to the corresponding INIT bit value when the associated inputs are applied. For instance, a Verilog INIT value of 64'h8000000000000000 (X"8000000000000000" for VHDL) will make the output zero unless all of the inputs are one (a 6-input AND gate). A Verilog INIT value of 64'hfffffffffffffffe (X"FFFFFFFFFFFFFFFE" for VHDL) will make the output one unless all zeros are on the inputs (a 6-input OR gate).

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

The Logic Table Method -A common method to determine the desired INIT value for a LUT is using a logic table. To do so, simply create a binary 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 is more self-documenting that the above method. However, this method does require the code to first specify the appropriate parameters.

Logic Table

Inputs						Outputs
I5	I4	I3	I2	I1	I0	LO
0	0	0	0	0	0	INIT[0]
0	0	0	0	0	1	INIT[1]

Inputs						Outputs
I5	**I4**	**I3**	**I2**	**I1**	**I0**	**LO**
0	0	0	0	1	0	INIT[2]
0	0	0	0	1	1	INIT[3]
0	0	0	1	0	0	INIT[4]
0	0	0	1	0	1	INIT[5]
0	0	0	1	1	0	INIT[6]
0	0	0	1	1	1	INIT[7]
0	0	1	0	0	0	INIT[8]
0	0	1	0	0	1	INIT[9]
0	0	1	0	1	0	INIT[10]
0	0	1	0	1	1	INIT[11]
0	0	1	1	0	0	INIT[12]
0	0	1	1	0	1	INIT[13]
0	0	1	1	1	0	INIT[14]
0	0	1	1	1	1	INIT[15]
0	1	0	0	0	0	INIT[16]
0	1	0	0	0	1	INIT[17]
0	1	0	0	1	0	INIT[18]
0	1	0	0	1	1	INIT[19]
0	1	0	1	0	0	INIT[20]
0	1	0	1	0	1	INIT[21]
0	1	0	1	1	0	INIT[22]
0	1	0	1	1	1	INIT[23]
0	1	1	0	0	0	INIT[24]
0	1	1	0	0	1	INIT[25]
0	1	1	0	1	0	INIT[26]
0	1	1	0	1	1	INIT[27]
0	1	1	1	0	0	INIT[28]
0	1	1	1	0	1	INIT[29]
0	1	1	1	1	0	INIT[30]
0	1	1	1	1	1	INIT[31]
1	0	0	0	0	0	INIT[32]
1	0	0	0	0	1	INIT[33]
1	0	0	0	1	0	INIT[34]
1	0	0	0	1	1	INIT[35]
1	0	0	1	0	0	INIT[36]
1	0	0	1	0	1	INIT[37]
1	0	0	1	1	0	INIT[38]

Inputs						Outputs
I5	**I4**	**I3**	**I2**	**I1**	**I0**	**LO**
1	0	0	1	1	1	INIT[39]
1	0	1	0	0	0	INIT[40]
1	0	1	0	0	1	INIT[41]
1	0	1	0	1	0	INIT[42]
1	0	1	0	1	1	INIT[43]
1	0	1	1	0	0	INIT[44]
1	0	1	1	0	1	INIT[45]
1	0	1	1	1	0	INIT[46]
1	0	1	1	1	1	INIT[47]
1	1	0	0	0	0	INIT[48]
1	1	0	0	0	1	INIT[49]
1	1	0	0	1	0	INIT[50]
1	1	0	0	1	1	INIT[51]
1	1	0	1	0	0	INIT[52]
1	1	0	1	0	1	INIT[53]
1	1	0	1	1	0	INIT[54]
1	1	0	1	1	1	INIT[55]
1	1	1	0	0	0	INIT[56]
1	1	1	0	0	1	INIT[57]
1	1	1	0	1	0	INIT[58]
1	1	1	0	1	1	INIT[59]
1	1	1	1	0	0	INIT[60]
1	1	1	1	0	1	INIT[61]
1	1	1	1	1	0	INIT[62]
1	1	1	1	1	1	INIT[63]
INIT = Binary equivalent of the hexadecimal number assigned to the INIT attribute						

Port Description

Name	Direction	Width	Function
LO	Output	1	6/5-LUT output or internal CLB connection
I0, I1, I2, I3, I4, I5	Input	1	LUT inputs

Design Entry Method

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

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
INIT	Hexadecimal	Any 64-Bit Value	All zeros	Specifies the logic value for the 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;


-- LUT6_L: 6-input Look-Up Table with local output
--         Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

LUT6_L_inst : LUT6_L
generic map (
   INIT => X"0000000000000000") -- Specify LUT Contents
port map (
   LO => LO, -- LUT local output
   I0 => I0, -- LUT input
   I1 => I1, -- LUT input
   I2 => I2, -- LUT input
   I3 => I3, -- LUT input
   I4 => I4, -- LUT input
   I5 => I5  -- LUT input
);

-- End of LUT6_L_inst instantiation
```

Verilog Instantiation Template

```
// LUT6_L: 6-input Look-Up Table with local output
//         Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

LUT6_L #(
   .INIT(64'h0000000000000000)  // Specify LUT Contents
) LUT6_L_inst (
   .LO(LO), // LUT local output
   .I0(I0), // LUT input
   .I1(I1), // LUT input
   .I2(I2), // LUT input
   .I3(I3), // LUT input
   .I4(I4), // LUT input
   .I5(I5)  // LUT input
);

// End of LUT6_L_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA Configurable Logic Block User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

MCB

Primitive: Memory Control Block

Introduction

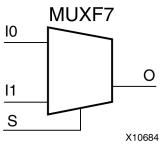
This element is a hardened memory controller supporting several popular memory interfaces. The MCB is only supported through the use of the Memory Interface Generator (MIG) tool. Details on the use and capability of the MCB can be found in the *Spartan-6 FPGA Memory Controller User Guide*, UG 388.

For More Information

- See the *Spartan-6 FPGA Memory Controller User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.
- See the *Spartan-6 FPGA User Documentation (User Guides and Data Sheets)*.

MUXF7

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



Introduction

This design element is a two input multiplexer for creating a function-of-7 look-up table or a 16-to-1 multiplexer in combination with two LUT6 look-up tables. Local outputs (LO) of two LUT6 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 O output is a general interconnect.

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

Logic Table

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

Port Descriptions

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

Design Entry Method

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

VHDL Instantiation Template

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

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


-- MUXF7: CLB MUX to tie two LUT6's together with general output
--        Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of MUXF7_inst instantiation
```

Verilog Instantiation Template

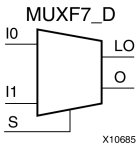
```
// MUXF7: CLB MUX to tie two LUT6's together with general output
//        Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

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

// End of MUXF7_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA Configurable Logic Block User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

MUXF7_D

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



Introduction

This design element is a two input multiplexer for creating a function-of-7 look-up table or a 16-to-1 multiplexer in combination with two LUT6 look-up tables. Local outputs (LO) of two LUT6 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.

See also MUXF7 and MUXF7_L.

Logic Table

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

Port Descriptions

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

Design Entry Method

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

VHDL Instantiation Template

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

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


-- MUXF7_D: CLB MUX to tie two LUT6's together with general and local outputs
--          Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

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 or LUT6 O6 pin)
   I1 => I1,  -- Input (tie to MUXF6 LO out or LUT6 O6 pin)
   S => S     -- Input select to MUX
);

-- End of MUXF7_D_inst instantiation
```

Verilog Instantiation Template

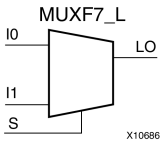
```
// MUXF7_D: CLB MUX to tie two LUT6's together with general and local outputs
//          Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

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 LUT6 O6 pin)
   .I1(I1),  // Input (tie to LUT6 O6 pin)
   .S(S)     // Input select to MUX
);

// End of MUXF7_D_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA Configurable Logic Block User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

MUXF7_L

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



Introduction

This design element is a two input multiplexer for creating a function-of-7 look-up table or a 16-to-1 multiplexer in combination with two LUT6 look-up tables. Local outputs (LO) of two LUT6 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.

See also MUXF7 and MUXF7_D.

Logic Table

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

Port Descriptions

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

Design Entry Method

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

VHDL Instantiation Template

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

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


-- MUXF7_L: CLB MUX to tie two LUT6's together with local output
--          Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

MUXF7_L_inst : MUXF7_L
port map (
   LO => LO,  -- Output of MUX to local routing
   I0 => I0,  -- Input (tie to MUXF6 LO out or LUT6 O6 pin)
   I1 => I1,  -- Input (tie to MUXF6 LO out or LUT6 O6 pin)
   S => S     -- Input select to MUX
);

-- End of MUXF7_L_inst instantiation
```

Verilog Instantiation Template

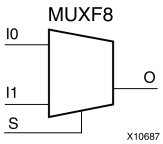
```
// MUXF7_L: CLB MUX to tie two LUT6's together with local output
//          Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

MUXF7_L MUXF7_L_inst (
   .LO(LO),  // Output of MUX to local routing
   .I0(I0),  // Input (tie to LUT6 O6 pin)
   .I1(I1),  // Input (tie to LUT6 O6 pin)
   .S(S)     // Input select to MUX
);

// End of MUXF7_L_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA Configurable Logic Block User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

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

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

Design Entry Method

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

VHDL Instantiation Template

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

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


-- MUXF8: CLB MUX to tie two MUXF7's together with general output
--        Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of MUXF8_inst instantiation
```

Verilog Instantiation Template

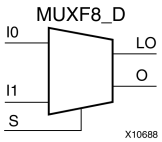
```
// MUXF8: CLB MUX to tie two MUXF7's together with general output
//        Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

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-6 FPGA Configurable Logic Block User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

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 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.

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

Logic Table

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

Port Descriptions

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

Design Entry Method

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

VHDL Instantiation Template

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

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


-- MUXF8_D: CLB MUX to tie two MUXF7's together with general and local outputs
--          Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

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

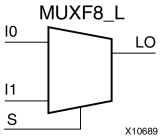
```
// MUXF8_D: CLB MUX to tie two MUXF7's together with general and local outputs
//          Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

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-6 FPGA Configurable Logic Block User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

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 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.

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

Logic Table

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

Port Descriptions

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

Design Entry Method

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

VHDL Instantiation Template

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

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


-- MUXF8_L: CLB MUX to tie two MUXF7's together with local output
--          Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

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

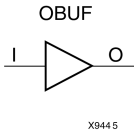
```
// MUXF8_L: CLB MUX to tie two MUXF7's together with local output
//          Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

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-6 FPGA Configurable Logic Block User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

OBUF

Primitive: Output Buffer



Introduction

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

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

Port Descriptions

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

Design Entry Method

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

Available Attributes

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

VHDL Instantiation Template

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

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


-- OBUF: Single-ended Output Buffer
--       Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of OBUF_inst instantiation
```

Verilog Instantiation Template

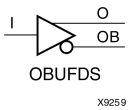
```
// OBUF: Single-ended Output Buffer
//       Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

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

// End of OBUF_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA SelectIO Resources User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

OBUFDS

Primitive: Differential Signaling Output Buffer



Introduction

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

Logic Table

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

Port Descriptions

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

Design Entry Method

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

Available Attributes

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

VHDL Instantiation Template

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

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


-- OBUFDS: Differential Output Buffer
--         Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of OBUFDS_inst instantiation
```

Verilog Instantiation Template

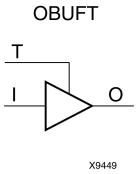
```
// OBUFDS: Differential Output Buffer
//         Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

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

// End of OBUFDS_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA SelectIO Resources User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

OBUFT

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



Introduction

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

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

Logic Table

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

Port Descriptions

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

Design Entry Method

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

Available Attributes

Attribute	Data 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. You should set this to the lowest setting tolerable for the design drive and timing requirements.
IOSTANDARD	String	See Data Sheet	"DEFAULT"	Assigns an I/O standard to the element.
SLEW	String	"SLOW" or "FAST"	"SLOW"	Specifies the slew rate of the output driver. See the Data Sheet for recommendations of the best setting for this attribute.

VHDL Instantiation Template

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

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


-- OBUFT: Single-ended 3-state Output Buffer
--        Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of OBUFT_inst instantiation
```

Verilog Instantiation Template

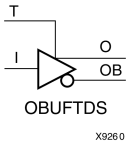
```
// OBUFT: Single-ended 3-state Output Buffer
//        Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

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

  // End of OBUFT_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA SelectIO Resources User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

OBUFTDS

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



Introduction

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

Logic Table

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

Port Descriptions

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

Design Entry Method

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

Available Attributes

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

VHDL Instantiation Template

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

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


-- OBUFTDS: Differential 3-state Output Buffer
--          Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of OBUFTDS_inst instantiation
```

Verilog Instantiation Template

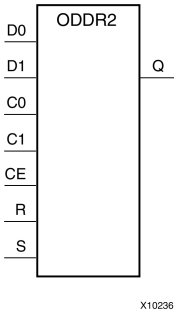
```
// OBUFTDS: Differential 3-state Output Buffer
//          Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

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

// End of OBUFTDS_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA SelectIO Resources User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

ODDR2

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



Introduction

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

Logic Table

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

Design Entry Method

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

Available Attributes

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

VHDL Instantiation Template

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

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


-- ODDR2: Output Double Data Rate Output Register with Set, Reset
--        and Clock Enable.
--        Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

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-6
// Xilinx HDL Libraries Guide, version 14.1

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)
```

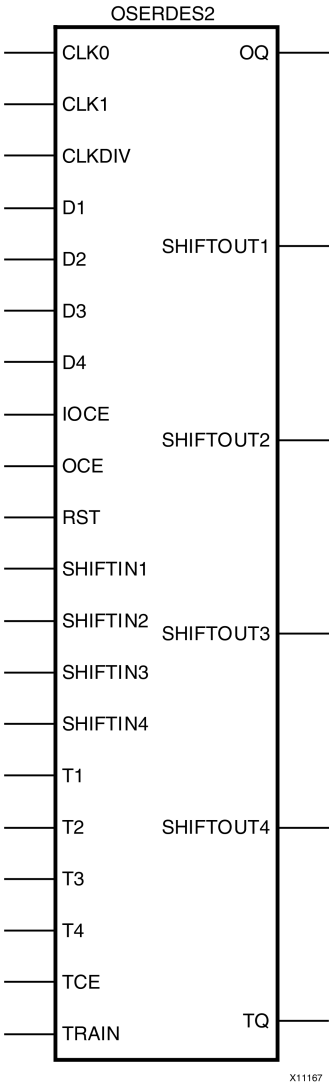
```
   .R(R),   // 1-bit reset input
   .S(S)    // 1-bit set input
);

// End of ODDR2_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA Configurable Logic Block User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

OSERDES2

Primitive: Dedicated IOB Output Serializer



Introduction

Each IOB contains a output serializer block that can be instantiated in a design by using the OSERDES2 primitive. OSERDES2 allows parallel-to-serial conversion with SerDes ratios of 2:1, 3:1, and 4:1. The SerDes ratio is the ratio between the high-speed I/O clock that istransmitting data, and the slower internal global clock used for processing the parallel data. For example, with an I/O clock running at 500 MHz to transmit data at 500 Mb/s, the OSERDES transfers four bits of data at one quarter of this rate (125 MHz) from the FPGA logic. When using differential outputs, the two OSERDES2 primitives associated with the two IOBs can be cascaded to allow higher SerDes ratios of 5:1, 6:1, 7:1 and 8:1.

Port Descriptions

Port	Direction	Width	Function
CLKDIV	Input	1	Global clock network input. This is the clock for the fabric domain.
CLK0	Input	1	I/O clock network input. Optionally invertible. This is the primary clock input used when the clock doubler circuit is not engaged. See DATA_RATE attribute for more information.
CLK1	Input	1	IO Clock network input. Optionally Invertible. This secondary clock input is only used when the clock doubler is engaged.
D1 - D4	Input	1	Parallel data inputs.
IOCE	Input	1	Data strobe signal derived from BUFIO2 CE. Strobes data capture to be correctly timed with respect to the I/O and global clocks for the SerDes mode selected.
OCE	Input	1	Clock enable for data inputs.
OQ	Output	1	Data path output to pad or IODELAY2.
RST	Input	1	Shared data, 3-state reset pin. Asynchronous only.
SHIFTIN1	Input	1	Cascade data input signal (dummy in Master). Used for DATA_WIDTHs greater than four.
SHIFTIN2	Input	1	Cascade 3-state input signal (dummy in master). Used for DATA_WIDTHs greater than 4.
SHIFTIN3	Input	1	Differential data input Signals (dummy in Slave).
SHIFTIN4	Input	1	Differential 3-state input signal (dummy in Slave)
SHIFTOUT1	Output	1	Cascade data output signal (dummy in Slave). Used for DATA_WIDTHs greater than four.
SHIFTOUT2	Output	1	Cascade 3-state output signal (dummy in Slave). Used for DATA_WIDTHs greater than 4.
SHIFTOUT3	Output	1	Differential data output signals (dummy in Master).
SHIFTOUT4	Output	1	Differential 3-state output signal (dummy in Master)
TCE	Input	1	Clock enable for 3-state inputs.
TQ	Output	1	3-state path output to pad or IODELAY2.
TRAIN	Input	1	Enable use of the training pattern. The train function is a means of specifying a fixed output pattern that can be used to calibrate the receiver of the signal. This port allows the FPGA logic to control whether the output is that fixed pattern or the input data from the pins.
T1 - T4	Input	1	3-state control inputs.

Design Entry Method

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

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
BYPASS_GCLK_FF	Boolean	FALSE, TRUE	FALSE	Bypass CLKDIV syncronization registers.
DATA_RATE_OQ	String	"DDR", "SDR"	"DDR"	Data rate setting. The DDR clock can be supplied by separate I/O clocks or by a single I/O clock. If two clocks are supplied, they must be approximately 180 out of phase.
DATA_RATE_OT	String"	"DDR", "BUF", "SDR"	"DDR"	3-state data rate setting. The DDR clock can be supplied by separate I/O clocks or by a single I/O clock. If two clocks are supplied, they must be approximately 180 out of phase.
DATA_WIDTH	Integer	2, 1, 3, 4, 5, 6, 7, 8	2	Data width. Determines the parallel data input width of the parallel-to-serial converter. Values greater than four are only valid when two OSERDES2 blocks are cascaded. In this case, the same value should be applied to both the master and slave blocks.
OUTPUT_MODE	String	"SINGLE_ENDED", "DIFFERENTIAL"	"SINGLE_ ENDED"	Output Mode.
SERDES_MODE	String	"MASTER", "SLAVE"	"MASTER"	Indicates whether OSERDES is being used alone, or as a Master or Slave when two OSERDES2 blocks are cascaded.
TRAIN_PATTERN	Integer	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15	0	Defines training pattern to be sent when TRAIN port is active.

VHDL Instantiation Template

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

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


-- OSERDES2: Output SERial/DESerializer
--           Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

OSERDES2_inst : OSERDES2
generic map (
   BYPASS_GCLK_FF => FALSE,       -- Bypass CLKDIV syncronization registers (TRUE/FALSE)
   DATA_RATE_OQ => "DDR",         -- Output Data Rate ("SDR" or "DDR")
   DATA_RATE_OT => "DDR",         -- 3-state Data Rate ("SDR" or "DDR")
   DATA_WIDTH => 2,               -- Parallel data width (2-8)
   OUTPUT_MODE => "SINGLE_ENDED", -- "SINGLE_ENDED" or "DIFFERENTIAL"
   SERDES_MODE => "NONE",         -- "NONE", "MASTER" or "SLAVE"
   TRAIN_PATTERN => 0             -- Training Pattern (0-15)
)
port map (
   OQ => OQ,               -- 1-bit output: Data output to pad or IODELAY2
   SHIFTOUT1 => SHIFTOUT1, -- 1-bit output: Cascade data output
   SHIFTOUT2 => SHIFTOUT2, -- 1-bit output: Cascade 3-state output
   SHIFTOUT3 => SHIFTOUT3, -- 1-bit output: Cascade differential data output
   SHIFTOUT4 => SHIFTOUT4, -- 1-bit output: Cascade differential 3-state output
   TQ => TQ,               -- 1-bit output: 3-state output to pad or IODELAY2
   CLK0 => CLK0,           -- 1-bit input: I/O clock input
   CLK1 => CLK1,           -- 1-bit input: Secondary I/O clock input
   CLKDIV => CLKDIV,       -- 1-bit input: Logic domain clock input
   -- D1 - D4: 1-bit (each) input: Parallel data inputs
   D1 => D1,
   D2 => D2,
   D3 => D3,
```

```
   D4 => D4,
   IOCE => IOCE,           -- 1-bit input: Data strobe input
   OCE => OCE,             -- 1-bit input: Clock enable input
   RST => RST,             -- 1-bit input: Asynchrnous reset input
   SHIFTIN1 => SHIFTIN1,   -- 1-bit input: Cascade data input
   SHIFTIN2 => SHIFTIN2,   -- 1-bit input: Cascade 3-state input
   SHIFTIN3 => SHIFTIN3,   -- 1-bit input: Cascade differential data input
   SHIFTIN4 => SHIFTIN4,   -- 1-bit input: Cascade differential 3-state input
   -- T1 - T4: 1-bit (each) input: 3-state control inputs
   T1 => T1,
   T2 => T2,
   T3 => T3,
   T4 => T4,
   TCE => TCE,             -- 1-bit input: 3-state clock enable input
   TRAIN => TRAIN          -- 1-bit input: Training pattern enable input
);

-- End of OSERDES2_inst instantiation
```

Verilog Instantiation Template

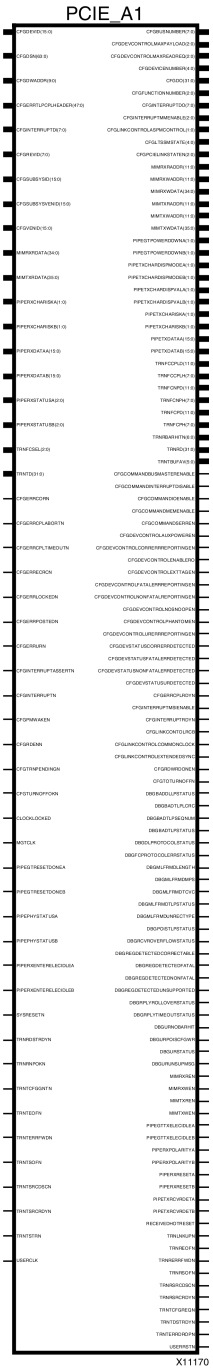
```
// OSERDES2: Output SERial/DESerializer
//           Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

OSERDES2 #(
   .BYPASS_GCLK_FF("FALSE"),     // Bypass CLKDIV syncronization registers (TRUE/FALSE)
   .DATA_RATE_OQ("DDR"),         // Output Data Rate ("SDR" or "DDR")
   .DATA_RATE_OT("DDR"),         // 3-state Data Rate ("SDR" or "DDR")
   .DATA_WIDTH(2),               // Parallel data width (2-8)
   .OUTPUT_MODE("SINGLE_ENDED"), // "SINGLE_ENDED" or "DIFFERENTIAL"
   .SERDES_MODE("NONE"),         // "NONE", "MASTER" or "SLAVE"
   .TRAIN_PATTERN(0)             // Training Pattern (0-15)
)
OSERDES2_inst (
   .OQ(OQ),               // 1-bit output: Data output to pad or IODELAY2
   .SHIFTOUT1(SHIFTOUT1), // 1-bit output: Cascade data output
   .SHIFTOUT2(SHIFTOUT2), // 1-bit output: Cascade 3-state output
   .SHIFTOUT3(SHIFTOUT3), // 1-bit output: Cascade differential data output
   .SHIFTOUT4(SHIFTOUT4), // 1-bit output: Cascade differential 3-state output
   .TQ(TQ),               // 1-bit output: 3-state output to pad or IODELAY2
   .CLK0(CLK0),           // 1-bit input: I/O clock input
   .CLK1(CLK1),           // 1-bit input: Secondary I/O clock input
   .CLKDIV(CLKDIV),       // 1-bit input: Logic domain clock input
   // D1 - D4: 1-bit (each) input: Parallel data inputs
   .D1(D1),
   .D2(D2),
   .D3(D3),
   .D4(D4),
   .IOCE(IOCE),           // 1-bit input: Data strobe input
   .OCE(OCE),             // 1-bit input: Clock enable input
   .RST(RST),             // 1-bit input: Asynchrnous reset input
   .SHIFTIN1(SHIFTIN1),   // 1-bit input: Cascade data input
   .SHIFTIN2(SHIFTIN2),   // 1-bit input: Cascade 3-state input
   .SHIFTIN3(SHIFTIN3),   // 1-bit input: Cascade differential data input
   .SHIFTIN4(SHIFTIN4),   // 1-bit input: Cascade differential 3-state input
   // T1 - T4: 1-bit (each) input: 3-state control inputs
   .T1(T1),
   .T2(T2),
   .T3(T3),
   .T4(T4),
   .TCE(TCE),             // 1-bit input: 3-state clock enable input
   .TRAIN(TRAIN)          // 1-bit input: Training pattern enable input
);

// End of OSERDES2_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA SelectIO Resources User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

PCIE_A1

Primitive: PCI Express



Introduction

This design element is intended for use in conjunction with other resources located in the FPGA, such as the RocketIO™ transceiver, block RAMs, and various clocking resources. To implement an PCI EXPRESS® design using PCIE_A1, designers must use the CORE Generator™ software tool (part of the ISE® Design Suite) to create a LogiCORE™ IP core for PCI EXPRESS designs. The LogiCORE IP instantiates the PCIE_A1 software primitive, connects the interfaces to the correct FPGA resources, sets all attributes, and presents a simple, user-friendly interface.

Design Entry Method

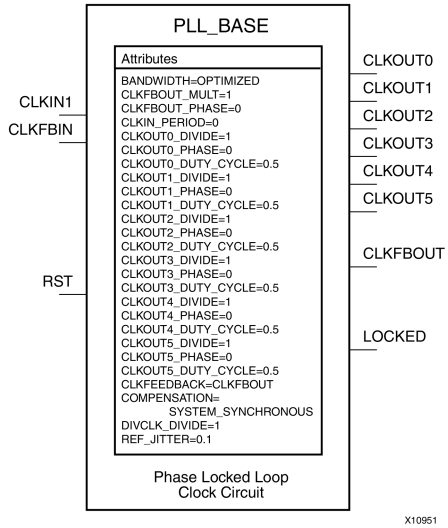
To instantiate this component, use the PCI EXPRESS® core or an associated core containing the component. Xilinx does not recommend direct instantiation of this component.

For More Information

- See the *Spartan-6 FPGA RocketIO GTP Transceivers User Guide*.
- See the *LogiCORE™ IP Spartan-6 FPGA Integrated Endpoint Block v1.1 for PCI EXPRESS® User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

PLL_BASE

Primitive: Basic Phase Locked Loop Clock Circuit



Introduction

This design element is a direct sub-set of the PLL_ADV design element, an embedded Phase Locked Loop clock circuit that provides added capabilities for clock synthesis and management both within the FPGA and in circuits external to the FPGA. The PLL_BASE is provided in order to ease the integration for most PLL clocking circuits. However, this primitive does not contain all of the functionality that the PLL can possibly provide. This component allows the input clock to be phase shifted, multiplied and divided, and supports other features, such as modification of the duty cycle and jitter filtering.

Port Descriptions

Port	Direction	Width	Function
Clock Outputs/Inputs			
CLKOUT0-5	Output	1	One of six phase controlled output clocks from the PLL.
CLKFBOUT	Output	1	Dedicated PLL feedback output used to determine how the PLL compensates clock network delay. Depending on the type of compensation desired, this output might or might not need to be connected.
CLKIN	Input	1	Clock source input to the PLL. This pin can be driven by a dedicated clock pin to the FPGA, a DCM output clock pin, or a BUFG output.
CLKFBIN	Input	1	Clock feedback input. This pin should only be sourced from the CLKFBOUT port.
Status Outputs/Control Inputs			
LOCKED	Output	1	Asynchronous output from the PLL that provides you with an indication the PLL has achieved phase alignment and is ready for operation.
RST	Input	1	Asynchronous reset of the PLL.

Design Entry Method

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

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
COMPENSATION	String	"SYSTEM_ SYNCHRONOUS", "SOURCE_ SYNCHRONOUS"	"SYSTEM_ SYNCHRONOUS"	Specifies the PLL phase compensation for the incoming clock.SYSTEM_SYNCHRONOUS attempts to compensate all clock delay while SOURCE_SYNCHRONOUS is used when a clock is provided with data and thus phased with the clock.
BANDWIDTH	String	"HIGH", "LOW", "OPTIMIZED"	"OPTIMIZED"	Specifies the PLL programming algorithm affecting the jitter, phase margin and other characteristics of the PLL.
CLKOUT0_DIVIDE, CLKOUT1_DIVIDE, CLKOUT2_DIVIDE, CLKOUT3_DIVIDE, CLKOUT4_DIVIDE, CLKOUT5_DIVIDE	Integer	1 to 128	1	Specifies the amount to divide the associated CLKOUT clock output if a different frequency is desired. This number in combination with the FBCLKOUT_MULT value determines the output frequency.
CLKOUT0_PHASE, CLKOUT1_PHASE, CLKOUT2_PHASE, CLKOUT3_PHASE, CLKOUT4_PHASE, CLKOUT5_PHASE	Real	0.01 to 360.0	0.0	Allows specification of the output phase relationship of the associated CLKOUT clock output in number of degrees offset (for instance, 90 indicates a 90 degree offset or 1/4 cycle phase offset while 180 indicates a 180 degree offset or 1/2 cycle phase offset).
CLKOUT0_DUTY_ CYCLE, CLKOUT1_DUTY_ CYCLE, CLKOUT2_DUTY_ CYCLE, CLKOUT3_DUTY_ CYCLE, CLKOUT4_DUTY_ CYCLE, CLKOUT5_DUTY_ CYCLE,	Real	0.01 to 0.99	0.50	Specifies the Duty Cycle of the associated CLKOUT clock output in percentage (i.e. 0.50 generates a 50% duty cycle).
CLKFBOUT_MULT	Integer	1 to 64	1	Specifies the amount to multiply all CLKOUT clock outputs if a different frequency is desired. This number in combination with the associated CLKOUT#_DIVIDE value determines the output frequency.
DIVCLK_DIVIDE	Integer	1 to 52	1	Specifies the division ratio for all output clocks.

Attribute	Data Type	Allowed Values	Default	Description
CLKFBOUT_PHASE	Real	0.0 to 360	0.0	Specifies the phase offset in degrees of the clock feedback output.
CLK_FEEDBACK	String	"CLKFBOUT", "CLKOUT0"	"CLKFBOUT"	Specifies the clock source to drive CLKFB_IN. Refer to *Spartan®-6 FPGA Clocking Resources User Guiden (UG382)* for correct usage of feedback resources and calculating VCO frequency.
REF_JITTER	Real	0.000 to 0.999	0.100	The reference clock jitter is specified in terms of the UI which is a percentage of the reference clock. The number provided should be the maximum peak to peak value on the input clock.
CLKIN_PERIOD	Real	1.000 to 52.630	None	Specified the input period in ns to the PLL CLKIN input.

VHDL Instantiation Template

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

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


-- PLL_BASE: Phase Locked Loop (PLL) Clock Management Component
--           Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

PLL_BASE_inst : PLL_BASE
generic map (
   BANDWIDTH => "OPTIMIZED",             -- "HIGH", "LOW" or "OPTIMIZED"
   CLKFBOUT_MULT => 1,                   -- Multiply value for all CLKOUT clock outputs (1-64)
   CLKFBOUT_PHASE => 0.0,                -- Phase offset in degrees of the clock feedback output
                                         -- (0.0-360.0).
   CLKIN_PERIOD => 0.0,                  -- Input clock period in ns to ps resolution (i.e. 33.333 is 30
                                         -- MHz).
   -- CLKOUT0_DIVIDE - CLKOUT5_DIVIDE: Divide amount for CLKOUT# clock output (1-128)
   CLKOUT0_DIVIDE => 1,
   CLKOUT1_DIVIDE => 1,
   CLKOUT2_DIVIDE => 1,
   CLKOUT3_DIVIDE => 1,
   CLKOUT4_DIVIDE => 1,
   CLKOUT5_DIVIDE => 1,
   -- CLKOUT0_DUTY_CYCLE - CLKOUT5_DUTY_CYCLE: Duty cycle for CLKOUT# clock output (0.01-0.99).
   CLKOUT0_DUTY_CYCLE => 0.5,
   CLKOUT1_DUTY_CYCLE => 0.5,
   CLKOUT2_DUTY_CYCLE => 0.5,
   CLKOUT3_DUTY_CYCLE => 0.5,
   CLKOUT4_DUTY_CYCLE => 0.5,
   CLKOUT5_DUTY_CYCLE => 0.5,
   -- CLKOUT0_PHASE - CLKOUT5_PHASE: Output phase relationship for CLKOUT# clock output (-360.0-360.0).
   CLKOUT0_PHASE => 0.0,
   CLKOUT1_PHASE => 0.0,
   CLKOUT2_PHASE => 0.0,
   CLKOUT3_PHASE => 0.0,
   CLKOUT4_PHASE => 0.0,
   CLKOUT5_PHASE => 0.0,
   CLK_FEEDBACK => "CLKFBOUT",           -- Clock source to drive CLKFBIN ("CLKFBOUT" or "CLKOUT0")
   COMPENSATION => "SYSTEM_SYNCHRONOUS", -- "SYSTEM_SYNCHRONOUS", "SOURCE_SYNCHRONOUS", "EXTERNAL"
   DIVCLK_DIVIDE => 1,                   -- Division value for all output clocks (1-52)
   REF_JITTER => 0.1,                    -- Reference Clock Jitter in UI (0.000-0.999).
   RESET_ON_LOSS_OF_LOCK => FALSE        -- Must be set to FALSE
)
port map (
   CLKFBOUT => CLKFBOUT, -- 1-bit output: PLL_BASE feedback output
```

```
   -- CLKOUT0 - CLKOUT5: 1-bit (each) output: Clock outputs
   CLKOUT0 => CLKOUT0,
   CLKOUT1 => CLKOUT1,
   CLKOUT2 => CLKOUT2,
   CLKOUT3 => CLKOUT3,
   CLKOUT4 => CLKOUT4,
   CLKOUT5 => CLKOUT5,
   LOCKED => LOCKED,     -- 1-bit output: PLL_BASE lock status output
   CLKFBIN => CLKFBIN,   -- 1-bit input: Feedback clock input
   CLKIN => CLKIN,       -- 1-bit input: Clock input
   RST => RST            -- 1-bit input: Reset input
);

-- End of PLL_BASE_inst instantiation
```

Verilog Instantiation Template

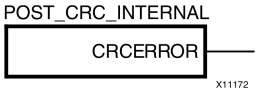
```
// PLL_BASE: Phase Locked Loop (PLL) Clock Management Component
//           Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

PLL_BASE #(
   .BANDWIDTH("OPTIMIZED"),             // "HIGH", "LOW" or "OPTIMIZED"
   .CLKFBOUT_MULT(1),                   // Multiply value for all CLKOUT clock outputs (1-64)
   .CLKFBOUT_PHASE(0.0),                // Phase offset in degrees of the clock feedback output (0.0-360.0).
   .CLKIN_PERIOD(0.0),                  // Input clock period in ns to ps resolution (i.e. 33.333 is 30
                                        // MHz).
   // CLKOUT0_DIVIDE - CLKOUT5_DIVIDE: Divide amount for CLKOUT# clock output (1-128)
   .CLKOUT0_DIVIDE(1),
   .CLKOUT1_DIVIDE(1),
   .CLKOUT2_DIVIDE(1),
   .CLKOUT3_DIVIDE(1),
   .CLKOUT4_DIVIDE(1),
   .CLKOUT5_DIVIDE(1),
   // CLKOUT0_DUTY_CYCLE - CLKOUT5_DUTY_CYCLE: Duty cycle for CLKOUT# clock output (0.01-0.99).
   .CLKOUT0_DUTY_CYCLE(0.5),
   .CLKOUT1_DUTY_CYCLE(0.5),
   .CLKOUT2_DUTY_CYCLE(0.5),
   .CLKOUT3_DUTY_CYCLE(0.5),
   .CLKOUT4_DUTY_CYCLE(0.5),
   .CLKOUT5_DUTY_CYCLE(0.5),
   // CLKOUT0_PHASE - CLKOUT5_PHASE: Output phase relationship for CLKOUT# clock output (-360.0-360.0).
   .CLKOUT0_PHASE(0.0),
   .CLKOUT1_PHASE(0.0),
   .CLKOUT2_PHASE(0.0),
   .CLKOUT3_PHASE(0.0),
   .CLKOUT4_PHASE(0.0),
   .CLKOUT5_PHASE(0.0),
   .CLK_FEEDBACK("CLKFBOUT"),           // Clock source to drive CLKFBIN ("CLKFBOUT" or "CLKOUT0")
   .COMPENSATION("SYSTEM_SYNCHRONOUS"), // "SYSTEM_SYNCHRONOUS", "SOURCE_SYNCHRONOUS", "EXTERNAL"
   .DIVCLK_DIVIDE(1),                   // Division value for all output clocks (1-52)
   .REF_JITTER(0.1),                    // Reference Clock Jitter in UI (0.000-0.999).
   .RESET_ON_LOSS_OF_LOCK("FALSE")      // Must be set to FALSE
)
PLL_BASE_inst (
   .CLKFBOUT(CLKFBOUT), // 1-bit output: PLL_BASE feedback output
   // CLKOUT0 - CLKOUT5: 1-bit (each) output: Clock outputs
   .CLKOUT0(CLKOUT0),
   .CLKOUT1(CLKOUT1),
   .CLKOUT2(CLKOUT2),
   .CLKOUT3(CLKOUT3),
   .CLKOUT4(CLKOUT4),
   .CLKOUT5(CLKOUT5),
   .LOCKED(LOCKED),     // 1-bit output: PLL_BASE lock status output
   .CLKFBIN(CLKFBIN),   // 1-bit input: Feedback clock input
   .CLKIN(CLKIN),       // 1-bit input: Clock input
   .RST(RST)            // 1-bit input: Reset input
);

// End of PLL_BASE_inst instantiation
```

For More Information

See the *Spartan-6 FPGA User Documentation (User Guides and Data Sheets)*.

POST_CRC_INTERNAL

Primitive: Post-configuration CRC error detection



Introduction

This primitive provides fabric access to post CRC error. This new primitive is added to provide more flexibility of POST_CRC usage. It is also the only access to POST CRC status when CRC_EXTSTAT_DISABLE is activated.

Port Descriptions

Port	Direction	Width	Function
CRCERROR	Output	1	Post-configuration CRC error

Design Entry Method

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

VHDL Instantiation Template

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

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


-- POST_CRC_INTERNAL: Post-configuration CRC error detection
--                    Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

POST_CRC_INTERNAL_inst : POST_CRC_INTERNAL
port map (
   CRCERROR => CRCERROR  -- 1-bit output: Post-configuration CRC error output
);

-- End of POST_CRC_INTERNAL_inst instantiation
```

Verilog Instantiation Template

```
// POST_CRC_INTERNAL: Post-configuration CRC error detection
//                    Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

POST_CRC_INTERNAL POST_CRC_INTERNAL_inst (
   .CRCERROR(CRCERROR)  // 1-bit output: Post-configuration CRC error output
);

// End of POST_CRC_INTERNAL_inst instantiation
```

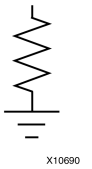
For More Information

- See the *Spartan-6 FPGA Configuration User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

PULLDOWN

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

PULLDOWN



Introduction

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

Port Descriptions

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

Design Entry Method

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

VHDL Instantiation Template

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

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


-- PULLDOWN: I/O Buffer Weak Pull-down
--           Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of PULLDOWN_inst instantiation
```

Verilog Instantiation Template

```
// PULLDOWN: I/O Buffer Weak Pull-down
//           Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

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

// End of PULLDOWN_inst instantiation
```

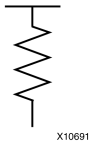
For More Information

- See the *Spartan-6 FPGA SelectIO Resources User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

PULLUP

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

PULLUP



Introduction

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

Port Descriptions

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

Design Entry Method

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

VHDL Instantiation Template

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

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


-- PULLUP: I/O Buffer Weak Pull-up
--         Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of PULLUP_inst instantiation
```

Verilog Instantiation Template

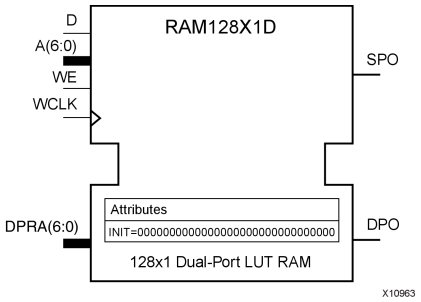
```
// PULLUP: I/O Buffer Weak Pull-up
//         Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

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

// End of PULLUP_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA SelectIO Resources User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

RAM128X1D

Primitive: 128-Deep by 1-Wide Dual Port Random Access Memory (Select RAM)



Introduction

This design element is a 128-bit deep by 1-bit wide random access memory and has a read/write port that writes the value on the D input data pin when the write enable (WE) is high to the location specified by the A address bus. This happens shortly after the rising edge of the WCLK and that same value is reflected in the data output SPO. When WE is low, an asynchronous read is initiated in which the contents of the memory location specified by the A address bus is output asynchronously to the SPO output. The read port can perform asynchronous read access of the memory by changing the value of the address bus DPRA, and by outputing that value to the DPO data output.

Port Descriptions

Port	Direction	Width	Function
SPO	Output	1	Read/Write port data output addressed by A
DPO	Output	1	Read port data output addressed by DPRA
D	Input	1	Write data input addressed by A
A	Input	7	Read/Write port address bus
DPRA	Input	7	Read port address bus
WE	Input	1	Write Enable
WCLK	Input	1	Write clock (reads are asynchronous)

If instantiated, the following connections should be made to this component:

- Tie the WCLK input to the desired clock source, the D input to the data source to be stored and the DPO output to an FDCE D input or other appropriate data destination.
- Optionally, the SPO output can also be connected to the appropriate data destination or else left unconnected.
- The WE clock enable pin should be connected to the proper write enable source in the design.
- The 7-bit A bus should be connected to the source for the read/write addressing and the 7-bit DPRA bus should be connected to the appropriate read address connections.
- An optional INIT attribute consisting of a 128-bit Hexadecimal value can be specified to indicate the initial contents of the RAM.

If left unspecified, the initial contents default to all zeros.

Design Entry Method

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

Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Hexadecimal	Any 128-Bit Value	All zeros	Specifies the 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;


-- RAM128X1D: 128-deep by 1-wide positive edge write, asynchronous read
--            dual-port distributed LUT RAM
--            Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

RAM128X1D_inst : RAM128X1D
generic map (
   INIT => X"00000000000000000000000000000000")
port map (
   DPO => DPO,      -- Read/Write port 1-bit ouput
   SPO => SPO,      -- Read port 1-bit output
   A => A,          -- Read/Write port 7-bit address input
   D => D,          -- RAM data input
   DPRA => DPRA,    -- Read port 7-bit address input
   WCLK => WCLK,    -- Write clock input
   WE => WE         -- RAM data input
);

-- End of RAM128X1D_inst instantiation
```

Verilog Instantiation Template

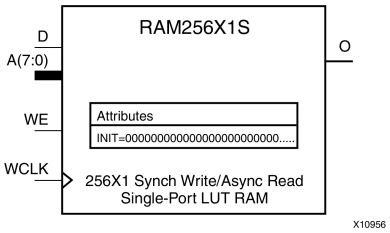
```
// RAM128X1D: 128-deep by 1-wide positive edge write, asynchronous read
//            dual-port distributed LUT RAM
//            Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

RAM128X1D #(
   .INIT(128'h00000000000000000000000000000000)
) RAM128X1D_inst (
   .DPO(DPO),   // Read port 1-bit output
   .SPO(SPO),   // Readw/rite port 1-bit output
   .A(A),       // Readw/rite port 7-bit address input
   .D(D),       // RAM data input
   .DPRA(DPRA), // Read port 7-bit address input
   .WCLK(WCLK), // Write clock input
   .WE(WE)      // Write enable input
);

// End of RAM128X1D_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA Configurable Logic Block User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

RAM256X1S

Primitive: 256-Deep by 1-Wide Random Access Memory (Select RAM)



Introduction

This design element is a 256-bit deep by 1-bit wide random access memory with synchronous write and asynchronous read capability. This RAM is implemented using the LUT resources of the device (also known as Select RAM), and does not consume any of the block RAM resources of the device. If a synchronous read capability is preferred, a register can be attached to the output and placed in the same slice as long as the same clock is used for both the RAM and the register. The RAM256X1S has an active, High write enable, WE, so that when that signal is High, and a rising edge occurs on the WCLK pin, a write is performed recording the value of the D input data pin into the memory array. The output O displays the contents of the memory location addressed by A, regardless of the WE value. When a write is performed, the output is updated to the new value shortly after the write completes.

Port Descriptions

Port	Direction	Width	Function
O	Output	1	Read/Write port data output addressed by A
D	Input	1	Write data input addressed by A
A	Input	8	Read/Write port address bus
WE	Input	1	Write Enable
WCLK	Input	1	Write clock (reads are asynchronous)

Design Entry Method

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

If instantiated, the following connections should be made to this component:

- Tie the WCLK input to the desired clock source, the D input to the data source to be stored, and the O output to an FDCE D input or other appropriate data destination.
- The WE clock enable pin should be connected to the proper write enable source in the design.
- The 8-bit A bus should be connected to the source for the read/write.
- An optional INIT attribute consisting of a 256-bit Hexadecimal value can be specified to indicate the initial contents of the RAM.

If left unspecified, the initial contents default to all zeros.

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
INIT	Hexadecimal	Any 256-Bit Value	All zeros	Specifies the 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;


-- RAM256X1S: 256-deep by 1-wide positive edge write, asynchronous read
--            single-port distributed LUT RAM
--            Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

RAM256X1S_inst : RAM256X1S
generic map (
   INIT => X"0000000000000000000000000000000000000000000000000000000000000000")
port map (
   O => O,  -- Read/Write port 1-bit ouput
   A => A,  -- Read/Write port 8-bit address input
   D => D,  -- RAM data input
   WCLK => WCLK,  -- Write clock input
   WE => WE -- Write enable input
);

-- End of RAM256X1S_inst instantiation
```

Verilog Instantiation Template

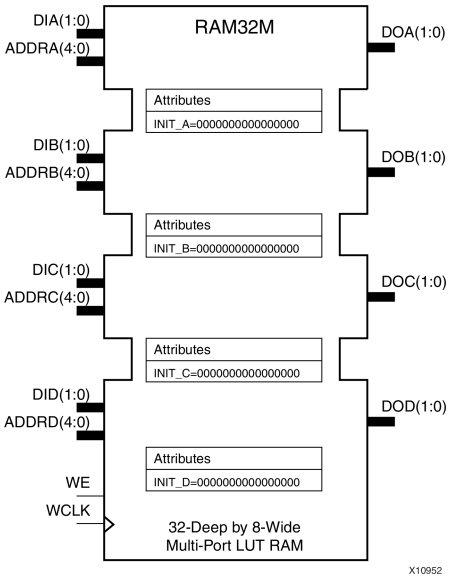
```
// RAM256X1S: 256-deep by 1-wide positive edge write, asynchronous read
//            single-port distributed LUT RAM
//            Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

RAM256X1S #(
   .INIT(256'h0000000000000000000000000000000000000000000000000000000000000000)
) RAM256X1S_inst (
   .O(O),       // Readw/rite port 1-bit output
   .A(A),       // Readw/rite port 8-bit address input
   .WE(WE),     // Write enable input
   .WCLK(WCLK), // Write clock input
   .D(D)        // RAM data input
);

// End of RAM256X1S_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA Configurable Logic Block User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

RAM32M

Primitive: 32-Deep by 8-bit Wide Multi Port Random Access Memory (Select RAM)



Introduction

This design element is a 32-bit deep by 8-bit wide, multi-port, random access memory with synchronous write and asynchronous independent, 2-bit, wide-read capability. This RAM is implemented using the LUT resources of the device known as SelectRAM™, and does not consume any of the Block RAM resources of the device. The RAM32M is implemented in a single slice and consists of one 8-bit write, 2-bit read port and three separate 2-bit read ports from the same memory. This configuration allows for byte-wide write and independent 2-bit read access RAM. If the DIA, DIB, DIC and DID inputs are all tied to the same data inputs, the RAM can become a 1 read/write port, 3 independent read port, 32x2 quad port memory. If DID is grounded, DOD is not used, while ADDRA, ADDRB and ADDRC are tied to the same address, the RAM becomes a 32x6 simple dual port RAM. If ADDRD is tied to ADDRA, ADDRB, and ADDRC, then the RAM is a 32x8 single port RAM. There are several other possible configurations for this RAM.

Port Descriptions

Port	Direction	Width	Function
DOA	Output	2	Read port data outputs addressed by ADDRA
DOB	Output	2	Read port data outputs addressed by ADDRB
DOC	Output	2	Read port data outputs addressed by ADDRC
DOD	Output	2	Read/Write port data outputs addressed by ADDRD
DIA	Input	2	Write data inputs addressed by ADDRD (read output is addressed by ADDRA)
DIB	Input	2	Write data inputs addressed by ADDRD (read output is addressed by ADDRB)
DIC	Input	2	Write data inputs addressed by ADDRD (read output is addressed by ADDRC)
DID	Input	2	Write data inputs addressed by ADDRD
ADDRA	Input	5	Read address bus A

Port	Direction	Width	Function
ADDRB	Input	5	Read address bus B
ADDRC	Input	5	Read address bus C
ADDRD	Input	5	8-bit data write port, 2-bit data read port address bus D
WE	Input	1	Write Enable
WCLK	Input	1	Write clock (reads are asynchronous)

Design Entry Method

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

This element can be inferred by some synthesis tools by describing a RAM with a synchronous write and asynchronous read capability. Consult your synthesis tool documentation for details on RAM inference capabilities and coding examples. Xilinx suggests that you instantiate RAM32Ms if you have a need to implicitly specify the RAM function, or if you need to manually place or relationally place the component. If a synchronous read capability is desired, the RAM32M outputs can be connected to an FDRSE (FDCPE is asynchronous set/reset is necessary) in order to improve the output timing of the function. However, this is not necessary for the proper operation of the RAM.

If you want to have the data clocked on the negative edge of a clock, an inverter can be described on the clock input to this component. This inverter will be absorbed into the block, giving you the ability to write to the RAM on falling clock edges.

If instantiated, the following connections should be made to this component. Tie the WCLK input to the desired clock source, the DIA, DIB, DIC and DID inputs to the data source to be stored and the DOA, DOB, DOC and DOD outputs to an FDCE D input or other appropriate data destination or left unconnected if not used. The WE clock enable pin should be connected to the proper write enable source in the design. The 5-bit ADDRD bus should be connected to the source for the read/write addressing and the 5-bit ADDRA, ADDRB and ADDRC buses should be connected to the appropriate read address connections. The optional INIT_A, INIT_B, INIT_C and INIT_D attributes consisting of a 64-bit hexadecimal values that specifies each port's initial memory contents can be specified. The INIT value correlates to the RAM addressing by the following equation: ADDRy[z] = INIT_y[2*z+1:2*z]. For instance, if the RAM ADDRC port is addressed to 00001, then the INIT_C[3:2] values would be the initial values shown on the DOC port before the first write occurs at that address. If left unspecified, the initial contents will default to all zeros.

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
INIT_A	Hexadecimal	Any 64-Bit Value	All zeros	Specifies the initial contents of the RAM on the A port.
INIT_B	Hexadecimal	Any 64-Bit Value	All zeros	Specifies the initial contents of the RAM on the B port.
INIT_C	Hexadecimal	Any 64-Bit Value	All zeros	Specifies the initial contents of the RAM on the C port.
INIT_D	Hexadecimal	Any 64-Bit Value	All zeros	Specifies the initial contents of the RAM on the D port.

VHDL Instantiation Template

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

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


-- RAM32M: 32-deep by 8-wide Multi Port LUT RAM
--         Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

RAM32M_inst : RAM32M
generic map (
   INIT_A => X"0000000000000000",   -- Initial contents of A port
   INIT_B => X"0000000000000000",   -- Initial contents of B port
   INIT_C => X"0000000000000000",   -- Initial contents of C port
   INIT_D => X"0000000000000000")    -- Initial contents of D port
port map (
   DOA => DOA, -- Read port A 2-bit output
   DOB => DOB, -- Read port B 2-bit output
   DOC => DOC, -- Read port C 2-bit output
   DOD => DOD, -- Read/Write port D 2-bit output
   ADDRA => ADDRA,   -- Read port A 5-bit address input
   ADDRB => ADDRB,   -- Read port B 5-bit address input
   ADDRC => ADDRC,   -- Read port C 5-bit address input
   ADDRD => ADDRD,   -- Read/Write port D 5-bit address input
   DIA => DIA, -- RAM 2-bit data write input addressed by ADDRD,
               -- read addressed by ADDRA
   DIB => DIB, -- RAM 2-bit data write input addressed by ADDRD,
               -- read addressed by ADDRB
   DIC => DIC, -- RAM 2-bit data write input addressed by ADDRD,
               -- read addressed by ADDRC
   DID => DID, -- RAM 2-bit data write input addressed by ADDRD,
               -- read addressed by ADDRD
   WCLK => WCLK,  -- Write clock input
   WE => WE       -- Write enable input
);
-- End of RAM32M_inst instantiation
```

Verilog Instantiation Template

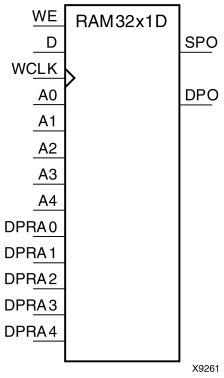
```
// RAM32M: 32-deep by 8-wide Multi Port LUT RAM
//         Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

RAM32M #(
   .INIT_A(64'h0000000000000000), // Initial contents of A Port
   .INIT_B(64'h0000000000000000), // Initial contents of B Port
   .INIT_C(64'h0000000000000000), // Initial contents of C Port
   .INIT_D(64'h0000000000000000)  // Initial contents of D Port
) RAM32M_inst (
   .DOA(DOA),     // Read port A 2-bit output
   .DOB(DOB),     // Read port B 2-bit output
   .DOC(DOC),     // Read port C 2-bit output
   .DOD(DOD),     // Readw/rite port D 2-bit output
   .ADDRA(ADDRA), // Read port A 5-bit address input
   .ADDRB(ADDRB), // Read port B 5-bit address input
   .ADDRC(ADDRC), // Read port C 5-bit address input
   .ADDRD(ADDRD), // Readw/rite port D 5-bit address input
   .DIA(DIA),     // RAM 2-bit data write input addressed by ADDRD,
                  //   read addressed by ADDRA
   .DIB(DIB),     // RAM 2-bit data write input addressed by ADDRD,
                  //   read addressed by ADDRB
   .DIC(DIC),     // RAM 2-bit data write input addressed by ADDRD,
                  //   read addressed by ADDRC
   .DID(DID),     // RAM 2-bit data write input addressed by ADDRD,
                  //   read addressed by ADDRD
   .WCLK(WCLK),   // Write clock input
   .WE(WE)        // Write enable input
);

// End of RAM32M_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA Configurable Logic Block User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

RAM32X1D

Primitive: 32-Deep by 1-Wide Static Dual Port Synchronous RAM



Introduction

The design element is a 32-word by 1-bit static dual port random access memory with synchronous write capability. The device has two separate address ports: the read address (DPRA4:DPRA0) and the write address (A4:A0). These two address ports are completely 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 5-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. You can initialize RAM32X1D during configuration using the INIT attribute. Mode selection is shown in the following logic table.

The SPO output reflects the data in the memory cell addressed by A4:A0. The DPO output reflects the data in the memory cell addressed by DPRA4:DPRA0. The write process is not affected by the address on the read address port.

Logic Table

Inputs			**Outputs**	
WE (Mode)	**WCLK**	**D**	**SPO**	**DPO**
0 (read)	X	X	data_a	data_d
1 (read)	0	X	data_a	data_d
1 (read)	1	X	data_a	data_d
1 (write)	↑	D	D	data_d
1 (read)	↓	X	data_a	data_d

Design Entry Method

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

Available Attributes

Attribute	Data Type	Allowed Values	Default	Descriptions
INIT	Hexadecimal	Any 32-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;


-- RAM32X1D: 32 x 1 positive edge write, asynchronous read
--              dual-port distributed RAM
--              Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

RAM32X1D_inst : RAM32X1D
generic map (
   INIT => X"00000000") -- Initial contents of RAM
port map (
   DPO => DPO,     -- Read-only 1-bit data output
   SPO => SPO,     -- R/W 1-bit data output
   A0 => A0,       -- R/W address[0] input bit
   A1 => A1,       -- R/W address[1] input bit
   A2 => A2,       -- R/W address[2] input bit
   A3 => A3,       -- R/W address[3] input bit
   A4 => A4,       -- R/W address[4] input bit
   D => D,         -- Write 1-bit data input
   DPRA0 => DPRA0, -- Read-only address[0] input bit
   DPRA1 => DPRA1, -- Read-only address[1] input bit
   DPRA2 => DPRA2, -- Read-only address[2] input bit
   DPRA3 => DPRA3, -- Read-only address[3] input bit
   DPRA4 => DPRA4, -- Read-only address[4] input bit
   WCLK => WCLK,   -- Write clock input
   WE => WE        -- Write enable input
);

-- End of RAM32X1D_inst instantiation
```

Verilog Instantiation Template

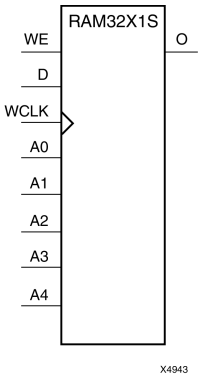
```
// RAM32X1D: 32 x 1 positive edge write, asynchronous read dual-port distributed RAM
//           Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

RAM32X1D #(
   .INIT(32'h00000000) // Initial contents of RAM
) RAM32X1D_inst (
   .DPO(DPO),     // Read-only 1-bit data output
   .SPO(SPO),     // Rw/ 1-bit data output
   .A0(A0),       // Rw/ address[0] input bit
   .A1(A1),       // Rw/ address[1] input bit
   .A2(A2),       // Rw/ address[2] input bit
   .A3(A3),       // Rw/ address[3] input bit
   .A4(A4),       // Rw/ address[4] input bit
   .D(D),         // Write 1-bit data input
   .DPRA0(DPRA0), // Read-only address[0] input bit
   .DPRA1(DPRA1), // Read-only address[1] input bit
   .DPRA2(DPRA2), // Read-only address[2] input bit
   .DPRA3(DPRA3), // Read-only address[3] input bit
   .DPRA4(DPRA4), // Read-only address[4] input bit
   .WCLK(WCLK),   // Write clock input
   .WE(WE)        // Write enable input
);

// End of RAM32X1D_inst instantiation
```

For More Information

See the *Spartan-6 FPGA User Documentation (User Guides and 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
CORE Generator™ and wizards	No
Macro support	No

Available Attributes

Attribute	**Data Type**	**Allowed Values**	**Default**	**Descriptions**
INIT	Hexadecimal	Any 32-Bit Value	All zeros	Specifies initial contents of the RAM.

VHDL Instantiation Template

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

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


-- RAM32X1S: 32 x 1 posedge write distributed (LUT) RAM
--           Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

RAM32X1S_inst : RAM32X1S
generic map (
   INIT => X"00000000")
port map (
   O => O,       -- RAM output
   A0 => A0,     -- RAM address[0] input
   A1 => A1,     -- RAM address[1] input
   A2 => A2,     -- RAM address[2] input
   A3 => A3,     -- RAM address[3] input
   A4 => A4,     -- RAM address[4] input
   D => D,       -- RAM data input
   WCLK => WCLK, -- Write clock input
   WE => WE      -- Write enable input
);

-- End of RAM32X1S_inst instantiation
```

Verilog Instantiation Template

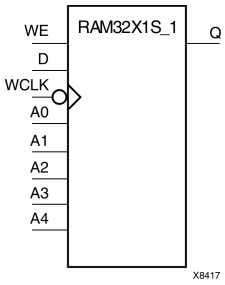
```
// RAM32X1S: 32 x 1 posedge write distributed (LUT) RAM
//           Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

RAM32X1S #(
   .INIT(32'h00000000)  // Initial contents of RAM
) RAM32X1S_inst (
   .O(O),       // RAM output
   .A0(A0),     // RAM address[0] input
   .A1(A1),     // RAM address[1] input
   .A2(A2),     // RAM address[2] input
   .A3(A3),     // RAM address[3] input
   .A4(A4),     // RAM address[4] input
   .D(D),       // RAM data input
   .WCLK(WCLK), // Write clock input
   .WE(WE)      // Write enable input
);

// End of RAM32X1S_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA Configurable Logic Block User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

RAM32X1S_1

Primitive: 32-Deep by 1-Wide Static Synchronous RAM with Negative-Edge Clock



Introduction

The design element is a 32-word by 1-bit static random access memory with synchronous write capability. When the write enable is Low, transitions on the write clock (WCLK) are ignored and data stored in the RAM is not affected. When (WE) is High, any negative transition on (WCLK) loads the data on the data input (D) into the word selected by the 5-bit address (A4:A0). For predictable performance, address and data inputs must be stable before a High-to-Low (WCLK) transition. This RAM block assumes an active-Low (WCLK). However, (WCLK) can be active-High or active-Low. Any inverter placed on the (WCLK) input net is absorbed into the block.

The signal output on the data output pin (O) is the data that is stored in the RAM at the location defined by the values on the address pins. You can initialize RAM32X1S_1 during configuration using the INIT attribute.

Logic Table

Inputs			**Outputs**
WE (Mode)	**WCLK**	**D**	**O**
0 (read)	X	X	Data
1 (read)	0	X	Data
1 (read)	1	X	Data
1 (write)	↓	D	D
1 (read)	↑	X	Data
Data = word addressed by bits A4:A0			

Design Entry Method

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

Available Attributes

Attribute	Data Type	Allowed Values	Default	Descriptions
INIT	Hexadecimal	Any 32-Bit Value	0	Initializes RAMs, registers, and look-up tables.

VHDL Instantiation Template

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

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


-- RAM32X1S_1: 32 x 1 negedge write distributed (LUT) RAM
--             Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

RAM32X1S_1_inst : RAM32X1S_1
generic map (
   INIT => X"00000000")
port map (
   O => O,       -- RAM output
   A0 => A0,     -- RAM address[0] input
   A1 => A1,     -- RAM address[1] input
   A2 => A2,     -- RAM address[2] input
   A3 => A3,     -- RAM address[3] input
   A4 => A4,     -- RAM address[4] input
   D => D,       -- RAM data input
   WCLK => WCLK, -- Write clock input
   WE => WE      -- Write enable input
);

-- End of RAM32X1S_1_inst instantiation
```

Verilog Instantiation Template

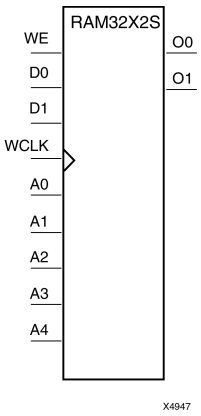
```
// RAM32X1S_1: 32 x 1 negedge write distributed (LUT) RAM
//             Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

RAM32X1S_1 #(
   .INIT(32'h00000000)  // Initial contents of RAM
)RAM32X1S_1_inst (
   .O(O),       // RAM output
   .A0(A0),     // RAM address[0] input
   .A1(A1),     // RAM address[1] input
   .A2(A2),     // RAM address[2] input
   .A3(A3),     // RAM address[3] input
   .A4(A4),     // RAM address[4] input
   .D(D),       // RAM data input
   .WCLK(WCLK), // Write clock input
   .WE(WE)      // Write enable input
);

// End of RAM32X1S_1_inst instantiation
```

For More Information

See the *Spartan-6 FPGA User Documentation (User Guides and Data Sheets)*.

RAM32X2S

Primitive: 32-Deep by 2-Wide Static Synchronous RAM



Introduction

The design element is a 32-word by 2-bit static random access memory with synchronous write capability. When the write enable (WE) is Low, transitions on the write clock (WCLK) are ignored and data stored in the RAM is not affected. When (WE) is High, any positive transition on (WCLK) loads the data on the data input (D1-D0) into the word selected by the 5-bit address (A4-A0). For predictable performance, address and data inputs must be stable before a Low-to-High (WCLK) transition. This RAM block assumes an active-High (WCLK). However, (WCLK) can be active-High or active-Low. Any inverter placed on the (WCLK) input net is absorbed into the block. The signal output on the data output pins (O1-O0) is the data that is stored in the RAM at the location defined by the values on the address pins.

You can use the INIT_00 and INIT_01 properties to specify the initial contents of RAM32X2S.

Logic Table

Inputs			**Outputs**
WE (Mode)	**WCLK**	**D**	**O0-O1**
0 (read)	X	X	Data
1 (read)	0	X	Data
1 (read)	1	X	Data
1 (write)	↑	D1:D0	D1:D0
1 (read)	↓	X	Data
Data = word addressed by bits A4:A0			

Design Entry Method

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

Available Attributes

Attribute	Data Type	Allowed Values	Default	Descriptions
INIT_00	Hexadecimal	Any 32-Bit Value	All zeros	INIT for bit 0 of RAM.
INIT_01	Hexadecimal	Any 32-Bit Value	All zeros	INIT for bit 1 of RAM.

VHDL Instantiation Template

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

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


-- RAM32X2S: 32 x 2 posedge write distributed (LUT) RAM
--           Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

RAM32X2S_inst : RAM32X2S
generic map (
   INIT_00 => X"00000000", --  INIT for bit 0 of RAM
   INIT_01 => X"00000000") --  INIT for bit 1 of RAM
port map (
   O0 => O0,     -- RAM data[0] output
   O1 => O1,     -- RAM data[1] output
   A0 => A0,     -- RAM address[0] input
   A1 => A1,     -- RAM address[1] input
   A2 => A2,     -- RAM address[2] input
   A3 => A3,     -- RAM address[3] input
   A4 => A4,     -- RAM address[4] input
   D0 => D0,     -- RAM data[0] input
   D1 => D1,     -- RAM data[1] input
   WCLK => WCLK, -- Write clock input
   WE => WE      -- Write enable input
);

-- End of RAM32X2S_inst instantiation
```

Verilog Instantiation Template

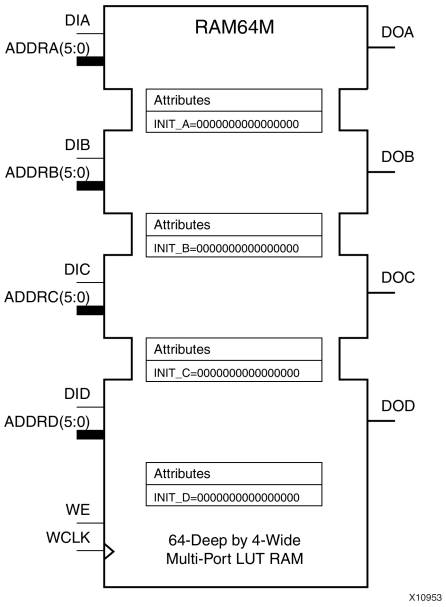
```
// RAM32X2S: 32 x 2 posedge write distributed (LUT) RAM
//           Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

RAM32X2S #(
   .INIT_00(32'h00000000), // INIT for bit 0 of RAM
   .INIT_01(32'h00000000)  // INIT for bit 1 of RAM
) RAM32X2S_inst (
   .O0(O0),     // RAM data[0] output
   .O1(O1),     // RAM data[1] output
   .A0(A0),     // RAM address[0] input
   .A1(A1),     // RAM address[1] input
   .A2(A2),     // RAM address[2] input
   .A3(A3),     // RAM address[3] input
   .A4(A4),     // RAM address[4] input
   .D0(D0),     // RAM data[0] input
   .D1(D1),     // RAM data[1] input
   .WCLK(WCLK), // Write clock input
   .WE(WE)      // Write enable input
);

// End of RAM32X2S_inst instantiation
```

For More Information

See the *Spartan-6 FPGA User Documentation (User Guides and Data Sheets)*.

RAM64M

Primitive: 64-Deep by 4-bit Wide Multi Port Random Access Memory (Select RAM)



Introduction

This design element is a 64-bit deep by 4-bit wide, multi-port, random access memory with synchronous write and asynchronous independent bit wide read capability. This RAM is implemented using the LUT resources of the device (also known as SelectRAM™) and does not consume any of the block RAM resources of the device. The RAM64M component is implemented in a single slice, and consists of one 4-bit write, 1-bit read port, and three separate 1-bit read ports from the same memory allowing for 4-bit write and independent bit read access RAM. If the DIA, DIB, DIC and DID inputs are all tied to the same data inputs, the RAM can become a 1 read/write port, 3 independent read port 64x1 quad port memory. If DID is grounded, DOD is not used. While ADDRA, ADDRB and ADDRC are tied to the same address the RAM becomes a 64x3 simple dual port RAM. If ADDRD is tied to ADDRA, ADDRB, and ADDRC; then the RAM is a 64x4 single port RAM. There are several other possible configurations for this RAM.

Port Descriptions

Port	Direction	Width	Function
DOA	Output	1	Read port data outputs addressed by ADDRA
DOB	Output	1	Read port data outputs addressed by ADDRB
DOC	Output	1	Read port data outputs addressed by ADDRC
DOD	Output	1	Read/Write port data outputs addressed by ADDRD
DIA	Input	1	Write data inputs addressed by ADDRD (read output is addressed by ADDRA)
DIB	Input	1	Write data inputs addressed by ADDRD (read output is addressed by ADDRB)
DIC	Input	1	Write data inputs addressed by ADDRD (read output is addressed by ADDRC)

Port	Direction	Width	Function
DID	Input	1	Write data inputs addressed by ADDRD
ADDRA	Input	6	Read address bus A
ADDRB	Input	6	Read address bus B
ADDRC	Input	6	Read address bus C
ADDRD	Input	6	4-bit data write port, 1-bit data read port address bus D
WE	Input	1	Write Enable
WCLK	Input	1	Write clock (reads are asynchronous)

Design Entry Method

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

This element can be inferred by some synthesis tools by describing a RAM with a synchronous write and asynchronous read capability. Consult your synthesis tool documentation for details on RAM inference capabilities and coding examples. Xilinx suggests that you instantiate RAM64Ms if you have a need to implicitly specify the RAM function, or if you need to manually place or relationally place the component. If a synchronous read capability is desired, the RAM64M outputs can be connected to an FDRSE (FDCPE is asynchronous set/reset is necessary) in order to improve the output timing of the function. However, this is not necessary for the proper operation of the RAM. If you want to have the data clocked on the negative edge of a clock, an inverter can be described on the clock input to this component. This inverter will be absorbed into the block giving the ability to write to the RAM on falling clock edges.

If instantiated, the following connections should be made to this component. Tie the WCLK input to the desired clock source, the DIA, DIB, DIC and DID inputs to the data source to be stored and the DOA, DOB, DOC and DOD outputs to an FDCE D input or other appropriate data destination or left unconnected if not used. The WE clock enable pin should be connected to the proper write enable source in the design. The 5-bit ADDRD bus should be connected to the source for the read/write addressing and the 5-bit ADDRA, ADDRB and ADDRC buses should be connected to the appropriate read address connections. The optional INIT_A, INIT_B, INIT_C and INIT_D attributes consisting of a 64-bit hexadecimal values that specifies each port's initial memory contents can be specified. The INIT value correlates to the RAM addressing by the following equation: ADDRy[z] = INIT_y[z].

For instance, if the RAM ADDRC port is addressed to 00001, then the INIT_C[1] values would be the initial values shown on the DOC port before the first write occurs at that address. If left unspecified, the initial contents will default to all zeros.

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
INIT_A	Hexadecimal	Any 64-Bit Value	All zero	Specifies the initial contents of the RAM on the A port.
INIT_B	Hexadecimal	Any 64-Bit Value	All zero	Specifies the initial contents of the RAM on the B port.
INIT_C	Hexadecimal	Any 64-Bit Value	All zero	Specifies the initial contents of the RAM on the C port.
INIT_D	Hexadecimal	Any 64-Bit Value	All zero	Specifies the initial contents of the RAM on the D port.

VHDL Instantiation Template

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

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


-- RAM64M: 64-deep by 4-wide Multi Port LUT RAM
--         Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

RAM64M_inst : RAM64M
generic map (
   INIT_A => X"0000000000000000",   -- Initial contents of A port
   INIT_B => X"0000000000000000",   -- Initial contents of B port
   INIT_C => X"0000000000000000",   -- Initial contents of C port
   INIT_D => X"0000000000000000")    -- Initial contents of D port
port map (
   DOA => DOA, -- Read port A 1-bit output
   DOB => DOB, -- Read port B 1-bit output
   DOC => DOC, -- Read port C 1-bit output
   DOD => DOD, -- Read/Write port D 1-bit output
   ADDRA => ADDRA,   -- Read port A 6-bit address input
   ADDRB => ADDRB,   -- Read port B 6-bit address input
   ADDRC => ADDRC,   -- Read port C 6-bit address input
   ADDRD => ADDRD,   -- Read/Write port D 6-bit address input
   DIA => DIA, -- RAM 1-bit data write input addressed by ADDRD,
               -- read addressed by ADDRA
   DIB => DIB, -- RAM 1-bit data write input addressed by ADDRD,
               -- read addressed by ADDRB
   DIC => DIC, -- RAM 1-bit data write input addressed by ADDRD,
               -- read addressed by ADDRC
   DID => DID, -- RAM 1-bit data write input addressed by ADDRD,
               -- read addressed by ADDRD
   WCLK => WCLK,  -- Write clock input
   WE => WE       -- Write enable input
);
-- End of RAM64M_inst instantiation
```

Verilog Instantiation Template

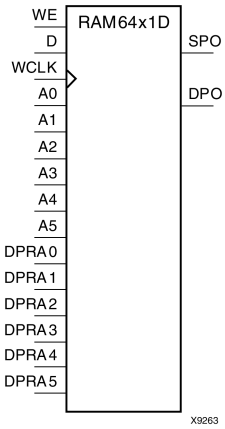
```
// RAM64M: 64-deep by 4-wide Multi Port LUT RAM
//         Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

RAM64M #(
   .INIT_A(64'h0000000000000000), // Initial contents of A Port
   .INIT_B(64'h0000000000000000), // Initial contents of B Port
   .INIT_C(64'h0000000000000000), // Initial contents of C Port
   .INIT_D(64'h0000000000000000)  // Initial contents of D Port
) RAM64M_inst (
   .DOA(DOA),     // Read port A 1-bit output
   .DOB(DOB),     // Read port B 1-bit output
   .DOC(DOC),     // Read port C 1-bit output
   .DOD(DOD),     // Readw/rite port D 1-bit output
   .DIA(DIA),     // RAM 1-bit data write input addressed by ADDRD,
                  //   read addressed by ADDRA
   .DIB(DIB),     // RAM 1-bit data write input addressed by ADDRD,
                  //   read addressed by ADDRB
   .DIC(DIC),     // RAM 1-bit data write input addressed by ADDRD,
                  //   read addressed by ADDRC
   .DID(DID),     // RAM 1-bit data write input addressed by ADDRD,
                  //   read addressed by ADDRD
   .ADDRA(ADDRA), // Read port A 6-bit address input
   .ADDRB(ADDRB), // Read port B 6-bit address input
   .ADDRC(ADDRC), // Read port C 6-bit address input
   .ADDRD(ADDRD), // Readw/rite port D 6-bit address input
   .WE(WE),       // Write enable input
   .WCLK(WCLK)    // Write clock input
);

// End of RAM64M_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA Configurable Logic Block User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

RAM64X1D

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



Introduction

This design element is a 64-word by 1-bit static dual port random access memory with synchronous write capability. The device has two separate address ports: the read address (DPRA5:DPRA0) and the write address (A5:A0). These two address ports are completely 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 6-bit (A0:A5) 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 A5:A0. The DPO output reflects the data in the memory cell addressed by DPRA5:DPRA0.

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

Logic Table

Inputs			Outputs	
WE (mode)	**WCLK**	**D**	**SPO**	**DPO**
0 (read)	X	X	data_a	data_d
1 (read)	0	X	data_a	data_d
1 (read)	1	X	data_a	data_d
1 (write)	↑	D	D	data_d
1 (read)	↓	X	data_a	data_d
data_a = word addressed by bits A5:A0 data_d = word addressed by bits DPRA5:DPRA0				

Design Entry Method

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

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
INIT	Hexadecimal	Any 64-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;


-- RAM64X1D: 64 x 1 negative edge write, asynchronous read
--           dual-port distributed RAM
--           Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

RAM64X1D_1_inst : RAM64X1D_1
generic map (
   INIT => X"0000000000000000") -- Initial contents of RAM
port map (
   DPO => DPO,     -- Read-only 1-bit data output
   SPO => SPO,     -- R/W 1-bit data output
   A0 => A0,       -- R/W address[0] input bit
   A1 => A1,       -- R/W address[1] input bit
   A2 => A2,       -- R/W address[2] input bit
   A3 => A3,       -- R/W address[3] input bit
   A4 => A4,       -- R/W address[4] input bit
   A5 => A5,       -- R/W address[5] input bit
   D => D,         -- Write 1-bit data input
   DPRA0 => DPRA0, -- Read-only address[0] input bit
   DPRA1 => DPRA1, -- Read-only address[1] input bit
   DPRA2 => DPRA2, -- Read-only address[2] input bit
   DPRA3 => DPRA3, -- Read-only address[3] input bit
   DPRA4 => DPRA4, -- Read-only address[4] input bit
   DPRA5 => DPRA5, -- Read-only address[5] input bit
   WCLK => WCLK,   -- Write clock input
   WE => WE        -- Write enable input
);

-- End of RAM64X1D_1_inst instantiation
```

Verilog Instantiation Template

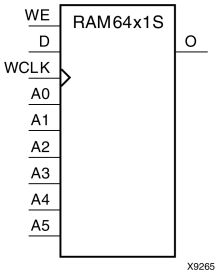
```
// RAM64X1D: 64 x 1 positive edge write, asynchronous read dual-port distributed RAM
//           Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

RAM64X1D #(
   .INIT(64'h0000000000000000) // Initial contents of RAM
) RAM64X1D_inst (
   .DPO(DPO),     // Read-only 1-bit data output
   .SPO(SPO),     // Rw/ 1-bit data output
   .A0(A0),       // Rw/ address[0] input bit
   .A1(A1),       // Rw/ address[1] input bit
   .A2(A2),       // Rw/ address[2] input bit
   .A3(A3),       // Rw/ address[3] input bit
   .A4(A4),       // Rw/ address[4] input bit
   .A5(A5),       // Rw/ address[5] input bit
   .D(D),         // Write 1-bit data input
   .DPRA0(DPRA0), // Read-only address[0] input bit
   .DPRA1(DPRA1), // Read-only address[1] input bit
   .DPRA2(DPRA2), // Read-only address[2] input bit
   .DPRA3(DPRA3), // Read-only address[3] input bit
   .DPRA4(DPRA4), // Read-only address[4] input bit
   .DPRA5(DPRA5), // Read-only address[5] input bit
   .WCLK(WCLK),   // Write clock input
   .WE(WE)        // Write enable input
);

// End of RAM64X1D_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA Configurable Logic Block User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

RAM64X1S

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



Introduction

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

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

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

Logic Table

Mode selection is shown in the following logic table

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

Design Entry Method

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

Available Attributes

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

VHDL Instantiation Template

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

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


-- RAM64X1S: 64 x 1 positive edge write, asynchronous read single-port distributed RAM
--           Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of RAM64X1S_inst instantiation
```

Verilog Instantiation Template

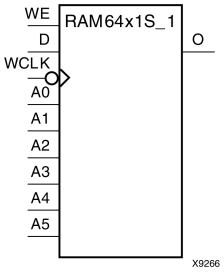
```
// RAM64X1S: 64 x 1 positive edge write, asynchronous read single-port distributed RAM
//           Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

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

// End of RAM64X1S_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA Configurable Logic Block User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

RAM64X1S_1

Primitive: 64-Deep by 1-Wide Static Synchronous RAM with Negative-Edge Clock



Introduction

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

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

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

Logic Table

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

Design Entry Method

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

Available Attributes

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

VHDL Instantiation Template

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

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


-- RAM64X1S_1: 64 x 1 negative edge write, asynchronous read single-port distributed RAM
--             Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

RAM64X1S_1_inst : RAM64X1S_1
generic map (
   INIT => X"0000000000000000")
port map (
   O => O,         -- 1-bit data output
   A0 => A0,       -- Address[0] input bit
   A1 => A1,       -- Address[1] input bit
   A2 => A2,       -- Address[2] input bit
   A3 => A3,       -- Address[3] input bit
   A4 => A4,       -- Address[4] input bit
   A5 => A5,       -- Address[5] input bit
   D => D,         -- 1-bit data input
   WCLK => WCLK,   -- Write clock input
   WE => WE        -- Write enable input
);

-- End of RAM64X1S_1_inst instantiation
```

Verilog Instantiation Template

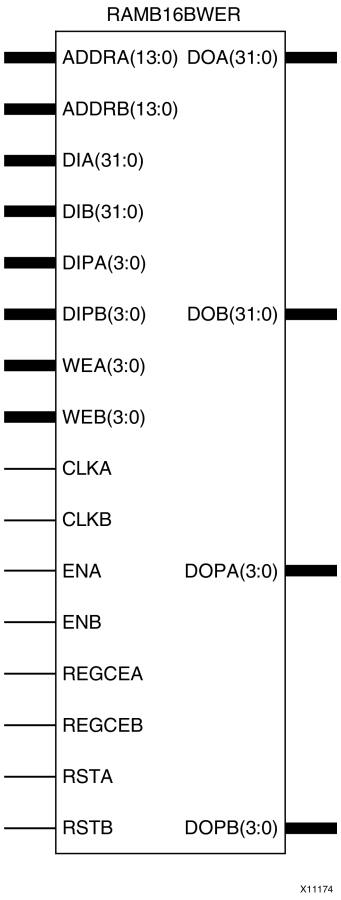
```
// RAM64X1S_1: 64 x 1 negative edge write, asynchronous read single-port distributed RAM
//             Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

RAM64X1S_1 #(
   .INIT(64'h0000000000000000) // Initial contents of RAM
) RAM64X1S_1_inst (
   .O(O),         // 1-bit data output
   .A0(A0),       // Address[0] input bit
   .A1(A1),       // Address[1] input bit
   .A2(A2),       // Address[2] input bit
   .A3(A3),       // Address[3] input bit
   .A4(A4),       // Address[4] input bit
   .A5(A5),       // Address[5] input bit
   .D(D),         // 1-bit data input
   .WCLK(WCLK),   // Write clock input
   .WE(WE)        // Write enable input
);

// End of RAM64X1S_1_inst instantiation
```

For More Information

See the *Spartan-6 FPGA User Documentation (User Guides and Data Sheets)*.

RAMB16BWER

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



Introduction

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 independently and asynchronously 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 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 value for either Port A or Port B.

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

Alternatively, the older 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 retarget them to a properly configured RAMB16BWER element.

Port	Direction	Width	Function
ADDRA[13:0]	Input	14	Port A address input bus. MSB always exists on ADDRA[13] while the LSB is determined by the settings for DATA_WIDTH_A.
ADDRB[13:0]	Input	14	Port B address input bus. MSB always exists on ADDRB[13] while the LSB is determined by the settings for DATA_WIDTH_B.
CLKA	Input	1	Port A clock input.
CLKB	Input	1	Port B clock input.
DIA[31:0]	Input	32	Port A data input bus.
DIB[31:0]	Input	32	Port B data input bus.
DIPA[3:0]	Input	4	Port A parity input bus.
DIPB[3:0]	Input	4	Port B parity input bus.
DOA[31:0]	Output	32	Port A data output bus.
DOB[31:0]	Output	32	Port B data output bus.
DOPA[3:0]	Output	4	Port A parity output bus.
DOPB[3:0]	Output	4	Port B parity output bus.
ENA	Input	1	Port A enable.
ENB	Input	1	Port B enable.
REGCEA	Input	1	Output register clock enable.
REGCEB	Input	1	Output register clock enable.
RSTA	Input	1	Port A output registers set/reset. This reset is configurable to be synchronous or asynchronous, depending on the value of the RSTTYPE attribute.
RSTB	Input	1	Port B output registers set/reset. This reset is configurable to be synchronous or asynchronous, depending on the value of the RSTTYPE attribute.
WEA[3:0]	Input	4	Port A byte-wide write enable.
WEB[3:0]	Input	4	Port B byte-wide write enable.

Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	Yes
Macro support	Yes

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 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	Data Type	Allowed Values	Default	Description
DATA_WIDTH_A	Integer	0, 1, 2, 4, 9, 18, 36	0	Specifies the configurable data width for port A. Need not equal the width for port B.
DATA_WIDTH_B	Integer	0, 1, 2, 4, 9, 18, 36	0	Specifies the configurable data width for port B. Need not equal the width for port A.
DOA_REG	Integer	0, 1	0	Set to 1 to use the A port output registers.
DOB_REG	Integer	0, 1	0	Set to 1 to use the B port output registers.
EN_RSTRAM_A	String	"TRUE", "FALSE"	"TRUE"	Disables A port RST capability when equal to FALSE and enables this capability when equal to TRUE.
EN_RSTRAM_B	String	"TRUE", "FALSE"	"TRUE"	Disables B port RST capability when equal to FALSE and enables this capability when equal to TRUE.
INIT_A	Hexa-decimal	36'h000000000 to 36'hfffffffff	All zeros	Specifies the initial value on the port A output after configuration.
INIT_B	Hexa-decimal	36'h000000000 to 36'hfffffffff	All zeros	Specifies the initial value on the Port B output after configuration.
INIT_FILE	String	String representing file name and location	NONE	File name of file used to specify initial RAM contents.
INIT_00 to INIT_3F	Hexa-decimal	Any 256 bit value	All zeros	Specifies the initial contents of the 16 kb data memory array.
INITP_01 to INITP_07	Hexa-decimal	Any 256 bit value	All zeros	Specifies the initial contents of the 2 kb parity data memory array.
RST_PRIORITY_A	String	"CE", "SR"	"CE"	When DOA_REG=0, selects the priority between the A port RAM EN and RST pin. When DOA_REG=1 (using the optional output register), selects priority between REGCE and RST pin.

Attribute	Data Type	Allowed Values	Default	Description
RST_PRIORITY_B	String	"CE", "SR"	"CE"	When DOB_REG=0, selects the priority between the B port RAM EN and RST pin. When DOB_REG=1 (using the optional output register), selects priority between REGCE and RST pin.
RSTTYPE	String	"SYNC", "ASYNC"	"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.
SIM_COLLISION_CHECK	String	"ALL", "GENERATE_X_ONLY", "WARNING_ONLY", "NONE"	"ALL"	Allows modification of the simulation behavior so that if a memory collision occurs: • "ALL" - Warning produced and affected outputs/memory go unknown (X). • "WARNING_ONLY" - Warning produced and affected outputs/memory retain last value. • "GENERATE_X_ONLY" - No warning, but 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	Hexa-decimal	36'h000000000 to 36'hfffffffff	All zeros	Specifies the output value of Port A upon the assertion of the reset (RSTA) signal.
SRVAL_B	Hexa-decimal	36'h000000000 to 36'hfffffffff	All zeros	Specifies the output value of Port B upon the assertion of the reset (RSTB) signal.
WRITE_MODE_A	String	"WRITE_FIRST", "READ_FIRST", "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.
WRITE_MODE_B	String	"WRITE_FIRST", "READ_FIRST", "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.

VHDL Instantiation Template

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

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


-- RAMB16BWER: 16k-bit Data and 2k-bit Parity Configurable Synchronous Dual Port Block RAM with Optional Output Registers
--             Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

RAMB16BWER_inst : RAMB16BWER
generic map (
   -- DATA_WIDTH_A/DATA_WIDTH_B: 0, 1, 2, 4, 9, 18, or 36
   DATA_WIDTH_A => 0,
   DATA_WIDTH_B => 0,
   -- DOA_REG/DOB_REG: Optional output register (0 or 1)
   DOA_REG => 0,
   DOB_REG => 0,
   -- EN_RSTRAM_A/EN_RSTRAM_B: Enable/disable RST
   EN_RSTRAM_A => TRUE,
   EN_RSTRAM_B => TRUE,
   -- INITP_00 to INITP_07: Initial memory contents.
   INITP_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_07 => X"0000000000000000000000000000000000000000000000000000000000000000",
   -- INIT_00 to INIT_3F: Initial memory contents.
   INIT_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_07 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_08 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_09 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_0A => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_0B => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_0C => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_0D => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_0E => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_0F => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_10 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_11 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_12 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_13 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_14 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_15 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_16 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_17 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_18 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_19 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_1A => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_1B => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_1C => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_1D => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_1E => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_1F => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_20 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_21 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_22 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_23 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_24 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_25 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_26 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_27 => X"0000000000000000000000000000000000000000000000000000000000000000",
```

```
   INIT_28 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_29 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_2A => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_2B => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_2C => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_2D => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_2E => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_2F => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_30 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_31 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_32 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_33 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_34 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_35 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_36 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_37 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_38 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_39 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_3A => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_3B => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_3C => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_3D => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_3E => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_3F => X"0000000000000000000000000000000000000000000000000000000000000000",
   -- INIT_A/INIT_B: Initial values on output port
   INIT_A => X"000000000",
   INIT_B => X"000000000",
   -- INIT_FILE: Optional file used to specify initial RAM contents
   INIT_FILE => "NONE",
   -- RSTTYPE: "SYNC" or "ASYNC"
   RSTTYPE => "SYNC",
   -- RST_PRIORITY_A/RST_PRIORITY_B: "CE" or "SR"
   RST_PRIORITY_A => "CE",
   RST_PRIORITY_B => "CE",
   -- SIM_COLLISION_CHECK: Collision check enable "ALL", "WARNING_ONLY", "GENERATE_X_ONLY" or "NONE"
   SIM_COLLISION_CHECK => "ALL",
   -- SIM_DEVICE: Must be set to "SPARTAN6" for proper simulation behavior
   SIM_DEVICE => "SPARTAN3ADSP",
   -- SRVAL_A/SRVAL_B: Set/Reset value for RAM output
   SRVAL_A => X"000000000",
   SRVAL_B => X"000000000",
   -- WRITE_MODE_A/WRITE_MODE_B: "WRITE_FIRST", "READ_FIRST", or "NO_CHANGE"
   WRITE_MODE_A => "WRITE_FIRST",
   WRITE_MODE_B => "WRITE_FIRST"
)
port map (
   -- Port A Data: 32-bit (each) output: Port A data
   DOA => DOA,       -- 32-bit output: A port data output
   DOPA => DOPA,     -- 4-bit output: A port parity output
   -- Port B Data: 32-bit (each) output: Port B data
   DOB => DOB,       -- 32-bit output: B port data output
   DOPB => DOPB,     -- 4-bit output: B port parity output
   -- Port A Address/Control Signals: 14-bit (each) input: Port A address and control signals
   ADDRA => ADDRA,   -- 14-bit input: A port address input
   CLKA => CLKA,     -- 1-bit input: A port clock input
   ENA => ENA,       -- 1-bit input: A port enable input
   REGCEA => REGCEA, -- 1-bit input: A port register clock enable input
   RSTA => RSTA,     -- 1-bit input: A port register set/reset input
   WEA => WEA,       -- 4-bit input: Port A byte-wide write enable input
   -- Port A Data: 32-bit (each) input: Port A data
   DIA => DIA,       -- 32-bit input: A port data input
   DIPA => DIPA,     -- 4-bit input: A port parity input
   -- Port B Address/Control Signals: 14-bit (each) input: Port B address and control signals
   ADDRB => ADDRB,   -- 14-bit input: B port address input
   CLKB => CLKB,     -- 1-bit input: B port clock input
   ENB => ENB,       -- 1-bit input: B port enable input
   REGCEB => REGCEB, -- 1-bit input: B port register clock enable input
   RSTB => RSTB,     -- 1-bit input: B port register set/reset input
   WEB => WEB,       -- 4-bit input: Port B byte-wide write enable input
   -- Port B Data: 32-bit (each) input: Port B data
   DIB => DIB,       -- 32-bit input: B port data input
   DIPB => DIPB      -- 4-bit input: B port parity input
);
```

```
-- End of RAMB16BWER_inst instantiation
```

Verilog Instantiation Template

```
// RAMB16BWER: 16k-bit Data and 2k-bit Parity Configurable Synchronous Dual Port Block RAM with Optional Output Registers
//             Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

RAMB16BWER #(
   // DATA_WIDTH_A/DATA_WIDTH_B: 0, 1, 2, 4, 9, 18, or 36
   .DATA_WIDTH_A(0),
   .DATA_WIDTH_B(0),
   // DOA_REG/DOB_REG: Optional output register (0 or 1)
   .DOA_REG(0),
   .DOB_REG(0),
   // EN_RSTRAM_A/EN_RSTRAM_B: Enable/disable RST
   .EN_RSTRAM_A("TRUE"),
   .EN_RSTRAM_B("TRUE"),
   // INITP_00 to INITP_07: Initial memory contents.
   .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),
   // INIT_00 to INIT_3F: Initial memory contents.
   .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),
```

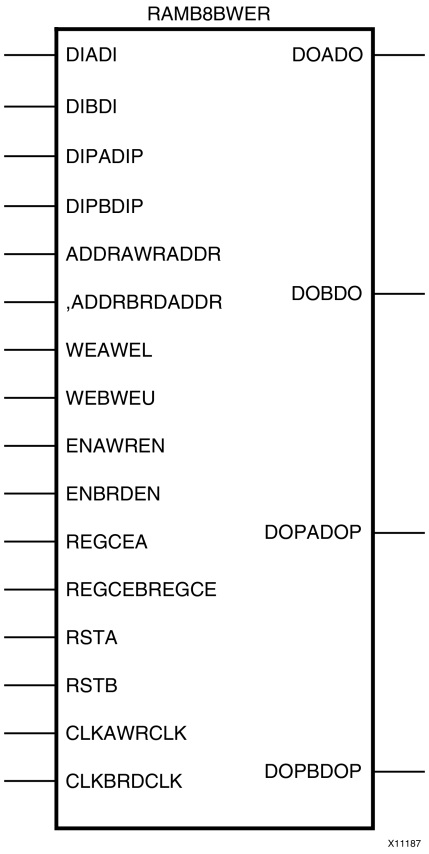
```
   .INIT_2A(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_2B(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_2C(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_2D(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .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),
   // INIT_A/INIT_B: Initial values on output port
   .INIT_A(36'h000000000),
   .INIT_B(36'h000000000),
   // INIT_FILE: Optional file used to specify initial RAM contents
   .INIT_FILE("NONE"),
   // RSTTYPE: "SYNC" or "ASYNC"
   .RSTTYPE("SYNC"),
   // RST_PRIORITY_A/RST_PRIORITY_B: "CE" or "SR"
   .RST_PRIORITY_A("CE"),
   .RST_PRIORITY_B("CE"),
   // SIM_COLLISION_CHECK: Collision check enable "ALL", "WARNING_ONLY", "GENERATE_X_ONLY" or "NONE"
   .SIM_COLLISION_CHECK("ALL"),
   // SIM_DEVICE: Must be set to "SPARTAN6" for proper simulation behavior
   .SIM_DEVICE("SPARTAN3ADSP"),
   // SRVAL_A/SRVAL_B: Set/Reset value for RAM output
   .SRVAL_A(36'h000000000),
   .SRVAL_B(36'h000000000),
   // WRITE_MODE_A/WRITE_MODE_B: "WRITE_FIRST", "READ_FIRST", or "NO_CHANGE"
   .WRITE_MODE_A("WRITE_FIRST"),
   .WRITE_MODE_B("WRITE_FIRST")
)
RAMB16BWER_inst (
   // Port A Data: 32-bit (each) output: Port A data
   .DOA(DOA),       // 32-bit output: A port data output
   .DOPA(DOPA),     // 4-bit output: A port parity output
   // Port B Data: 32-bit (each) output: Port B data
   .DOB(DOB),       // 32-bit output: B port data output
   .DOPB(DOPB),     // 4-bit output: B port parity output
   // Port A Address/Control Signals: 14-bit (each) input: Port A address and control signals
   .ADDRA(ADDRA),   // 14-bit input: A port address input
   .CLKA(CLKA),     // 1-bit input: A port clock input
   .ENA(ENA),       // 1-bit input: A port enable input
   .REGCEA(REGCEA), // 1-bit input: A port register clock enable input
   .RSTA(RSTA),     // 1-bit input: A port register set/reset input
   .WEA(WEA),       // 4-bit input: Port A byte-wide write enable input
   // Port A Data: 32-bit (each) input: Port A data
   .DIA(DIA),       // 32-bit input: A port data input
   .DIPA(DIPA),     // 4-bit input: A port parity input
   // Port B Address/Control Signals: 14-bit (each) input: Port B address and control signals
   .ADDRB(ADDRB),   // 14-bit input: B port address input
   .CLKB(CLKB),     // 1-bit input: B port clock input
   .ENB(ENB),       // 1-bit input: B port enable input
   .REGCEB(REGCEB), // 1-bit input: B port register clock enable input
   .RSTB(RSTB),     // 1-bit input: B port register set/reset input
   .WEB(WEB),       // 4-bit input: Port B byte-wide write enable input
   // Port B Data: 32-bit (each) input: Port B data
   .DIB(DIB),       // 32-bit input: B port data input
   .DIPB(DIPB)      // 4-bit input: B port parity input
);

// End of RAMB16BWER_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA Block RAM User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

RAMB8BWER

Primitive: 8K-bit Data and 1K-bit Parity Configurable Synchronous Dual Port Block RAM with Optional Output Registers



Introduction

Spartan®-6 devices contain several block RAM memories that can be configured as general-purpose RAM/ROM memories. These block RAM memories offer fast and flexible storage of large amounts of on-chip data. The RAMB8BWER allows access to the block RAM in the 8KB data + 1KB parity configuration. This element can be configured and used as a 1-bit wide by 8K deep to an 18-bit wide by 512-bit deep true dual port RAM. This element can also be configured as a 36-bit wide by 246 deep simple dual port RAM. Both read and write operations are fully synchronous to the supplied clock(s) to the component. However, the READ and WRITE ports can 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 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

Port	Direction	Width	Function
ADDRAWRADDR[12:0]	Input	13	Port A address input bus when RAM_MODE=TDP. MSB always exists on ADDRAWRADDR[12] while the LSB is determined by the settings for DATA_WIDTH_A. Write address input bus when RAM_MODE=SDP.
ADDRBRDADDR[12:0]	Input	13	Port B address input bus when RAM_MODE=TDP. MSB always exists on ADDRBWRADDR[12] while the LSB is determined by the settings for DATA_WIDTH_B. Write address input bus when RAM_MODE=SDP.
CLKAWRCLK	Input	1	Port A clock input/Write clock input.
CLKBRDCLK	Input	1	Port B clock input/Read clock input.
DIADI[15:0]	Input	16	Port A data input bus when RAM_MODE=TDP. Data input bus addressed by WRADDR when RAM_MODE=SDP. DIADI is the logical DI[15:0] for SDP mode.
DIBDI[15:0]	Input	16	Port B data input bus when RAM_MODE=TDP. Data input bus addressed by WRADDR when RAM_MODE=SDP. DIBDI is the logical DI[31:16] for SDP mode.
DIPADIP[1:0]	Input	2	Port A parity data input bus when RAM_MODE=TDP. Data parity input bus addressed by WRADDR when RAM_MODE=SDP. DIPADIP is the logical DIP[1:0] for SDP mode.
DIPBDIP[1:0]	Input	2	Port B parity data input bus when RAM_MODE=TDP. Data parity input bus addressed by WRADDR when RAM_MODE=SDP. DIPBDIP is the logical DIP[3:2] for SDP mode.
DOADO[15:0]	Output	16	Port A data output bus/Data output bus addressed by RDADDR. When RAM_MODE=SDP, DOADO is the logical DO[15:0].
DOBDO[15:0]	Output	16	Port B data output bus/Data output bus addressed by RDADDR. When RAM_MODE=SDP, DOBDO is the logical DO[31:16].
DOPADOP[1:0]	Output	2	Port A parity data output bus/Data parity output bus addressed by RDADDR. When RAM_MODE=SDP, DOPADOP is the logical DOP[1:0].
DOPBDOP[1:0]	Output	2	Port B parity data output bus/Data parity output bus addressed by RDADDR. When RAM_MODE=SDP, DOPBDOP is the logical DOP[3:2].
ENAWREN	Input	1	Port A RAM enable/Write enable.
ENBRDEN	Input	1	Port B RAM enable/Read enable.
REGCEA	Input	1	Port A output register clock enable input (valid only when DOA_REG=1). Not used when RAM_MODE=SDP.
REGCEBREGCE	Input	1	Port B output register clock enable input (valid only when DOB_REG=1). Output register clock enable input when RAM_MODE=SDP.
RSTA	Input	1	Port A set/reset to value indicated by SRVAL_A. This reset is configurable to be synchronous or asynchronous depending on the value of the RSTTYPE attribute. Affects the output value on the output registers (DOA_REG=1) as well as on the output latches. Not used when RAM_MODE=SDP.

Port	Direction	Width	Function
RSTBRST	Input	1	Port B set/reset to value indicated by SRVAL_B. This reset is configurable to be synchronous or asynchronous depending on the value of the RSTTYPE attribute. Affects the output value on the output registers (DOB_REG=1) as well as on the output latches. This is the Reset input when RAM_MODE=SDP.
WEAWEL	Input	2	Port A byte-wide write enable when RAM_MODE=TDP. In SDP mode, WEAWEL is logical WE[1:0].
WEBWEU	Input	2	Port B byte-wide write enable when RAM_MODE=TDP. In SDP mode, WEBWEU is logical WE[3:2].

Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	Yes
Macro support	Yes

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
DATA_WIDTH_A	Integer	0, 1, 2, 4, 9, 18, 36	0	Specifies the configurable data width for port A. Need not equal the width for port B. A width of 36 is valid for SDP mode only.
DATA_WIDTH_B	Integer	0, 1, 2, 4, 9, 18, 36	0	Specifies the configurable data width for port B. Need not equal the width for port A. A width of 36 is valid for SDP mode only.
DOA_REG	Integer	0, 1	0	Set to 1 to use the A port output registers. Applies to port A in TDP mode and up to 18 low order bits (including parity bits) in SDP mode.
DOB_REG	Integer	0, 1	0	Set to 1 to use the B port output registers. Applies to port B in TDP mode and up to 18 high order bits (including parity bits) in SDP mode.
EN_RSTRAM_A	String	"TRUE", "FALSE"	"TRUE"	Disables A port RST capability when equal to FALSE and enables this capability when equal to TRUE.
EN_RSTRAM_B	String	"TRUE", "FALSE"	"TRUE"	Disables B port RST capability when equal to FALSE and enables this capability when equal to TRUE.
INIT_A	Hexa-decimal	18'h00000 to 18'h3ffff	All zeros	Specifies the initial value on the port A output after configuration. Applies to port A in TDP mode and up to 18 low order bits (including parity bits) in SDP mode.
INIT_B	Hexa-decimal	18'h00000 to 18'h3ffff	All zeros	Specifies the initial value on the port B output after configuration. Applies to port B in TDP mode and up to 18 high order bits (including parity bits) in SDP mode.

Attribute	Data Type	Allowed Values	Default	Description
INIT_FILE	String	String representing file name and location	None	File name of file used to specify initial RAM contents.
INIT_00 to INIT_1F	Hexa-decimal	Any 256 bit value	All zeros	Allows specification of the initial contents of the 8KB data memory array.
INITP_01 to INITP_03	Hexa-decimal	Any 256 bit value	All zeros	Allows specification of the initial contents of the 1KB parity data memory array.
RAM_MODE	String	"TDP", "SDP"	"TDP"	Select "SDP" to configure this element as a simple dual port RAM (write-only on one port and read-only on the other). Select "TDP" to configure this element as a true dual port RAM (read and write capability on one or both ports).
RST_PRIORITY_A	String	"CE", "SR"	"CE"	When DOA_REG=0, selects the priority between the A port RAM EN and RST pin. When DOA_REG=1 (using the optional output register), selects priority between REGCE and RST pin.
RST_PRIORITY_B	String	"CE", "SR"	"CE"	When DOB_REG=0, selects the priority between the B port RAM EN and RST pin. When DOB_REG=1 (using the optional output register), selects priority between REGCE and RST pin.
RSTTYPE	String	"SYNC", "ASYNC"	"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.
SIM_COLLISION_CHECK	String	"ALL", "GENERATE_X_ONLY", "WARNING_ONLY", "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, but 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.

Attribute	Data Type	Allowed Values	Default	Description
SRVAL_A	Hexa-decimal	18'h00000 to 18'h3ffff	All zeros	Specifies the output value of Port A upon the assertion of the reset (RSTA) signal. Applies to port A in TDP mode. In SDP mode, use only SRVAL_A if the port width is 18 bits or less. If the port width is greater than 18 bits, SRVAL_A applies to the 18 low order bits (including parity bits).
SRVAL_B	Hexa-decimal	18'h00000 to 18'h3ffff	All zeros	Specifies the output value of Port B upon the assertion of the reset (RSTB) signal. Applies to port B in TDP mode. In SDP mode, use only SRVAL_A if the port width is 18 bits or less. If the port width is greater than 18 bits, SRVAL_B applies to the 18 high order bits (including parity bits).
WRITE_MODE_A	String	"WRITE_FIRST", "READ_FIRST", "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. When RAM_MODE=SDP, WRITE_MODE_A must equal "READ_FIRST" (when using a common clock on both ports) or "WRITE_FIRST" (when using different clocks on both ports).
WRITE_MODE_B	String	"WRITE_FIRST", "READ_FIRST", "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. When RAM_MODE=SDP, WRITE_MODE_B must equal "READ_FIRST" (when using a common clock on both ports) or "WRITE_FIRST" (when using different clocks on both ports).

VHDL Instantiation Template

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

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


-- RAMB8BWER: 8k-bit Data and 1k-bit Parity Configurable Synchronous Block RAM
```

```
--            Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

RAMB8BWER_inst : RAMB8BWER
generic map (
   -- DATA_WIDTH_A/DATA_WIDTH_B: 'If RAM_MODE="TDP": 0, 1, 2, 4, 9 or 18; If RAM_MODE="SDP": 36'
   DATA_WIDTH_A => 0,
   DATA_WIDTH_B => 0,
   -- DOA_REG/DOB_REG: Optional output register (0 or 1)
   DOA_REG => 0,
   DOB_REG => 0,
   -- EN_RSTRAM_A/EN_RSTRAM_B: Enable/disable RST
   EN_RSTRAM_A => TRUE,
   EN_RSTRAM_B => TRUE,
   -- INITP_00 to INITP_03: Initial memory contents.
   INITP_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
   -- INIT_00 to INIT_1F: Initial memory contents.
   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_A/INIT_B: Initial values on output port
   INIT_A => X"00000",
   INIT_B => X"00000",
   -- INIT_FILE: Not Supported
   INIT_FILE => "NONE",                                                    -- Do not modify
   -- RAM_MODE: "SDP" or "TDP"
   RAM_MODE => "TDP",
   -- RSTTYPE: "SYNC" or "ASYNC"
   RSTTYPE => "SYNC",
   -- RST_PRIORITY_A/RST_PRIORITY_B: "CE" or "SR"
   RST_PRIORITY_A => "CE",
   RST_PRIORITY_B => "CE",
   -- SIM_COLLISION_CHECK: Collision check enable "ALL", "WARNING_ONLY", "GENERATE_X_ONLY" or "NONE"
   SIM_COLLISION_CHECK => "ALL",
   -- SRVAL_A/SRVAL_B: Set/Reset value for RAM output
   SRVAL_A => X"00000",
   SRVAL_B => X"00000",
   -- WRITE_MODE_A/WRITE_MODE_B: "WRITE_FIRST", "READ_FIRST", or "NO_CHANGE"
   WRITE_MODE_A => "WRITE_FIRST",
   WRITE_MODE_B => "WRITE_FIRST"
)
```

```
port map (
   -- Port A Data: 16-bit (each) output: Port A data
   DOADO => DOADO,             -- 16-bit output: A port data/LSB data output
   DOPADOP => DOPADOP,         -- 2-bit output: A port parity/LSB parity output
   -- Port B Data: 16-bit (each) output: Port B data
   DOBDO => DOBDO,             -- 16-bit output: B port data/MSB data output
   DOPBDOP => DOPBDOP,         -- 2-bit output: B port parity/MSB parity output
   -- Port A Address/Control Signals: 13-bit (each) input: Port A address and control signals (write port
   -- when RAM_MODE="SDP")
   ADDRAWRADDR => ADDRAWRADDR, -- 13-bit input: A port address/Write address input
   CLKAWRCLK => CLKAWRCLK,     -- 1-bit input: A port clock/Write clock input
   ENAWREN => ENAWREN,         -- 1-bit input: A port enable/Write enable input
   REGCEA => REGCEA,           -- 1-bit input: A port register enable input
   RSTA => RSTA,               -- 1-bit input: A port set/reset input
   WEAWEL => WEAWEL,           -- 2-bit input: A port write enable input
   -- Port A Data: 16-bit (each) input: Port A data
   DIADI => DIADI,             -- 16-bit input: A port data/LSB data input
   DIPADIP => DIPADIP,         -- 2-bit input: A port parity/LSB parity input
   -- Port B Address/Control Signals: 13-bit (each) input: Port B address and control signals (read port
   -- when RAM_MODE="SDP")
   ADDRBRDADDR => ADDRBRDADDR, -- 13-bit input: B port address/Read address input
   CLKBRDCLK => CLKBRDCLK,     -- 1-bit input: B port clock/Read clock input
   ENBRDEN => ENBRDEN,         -- 1-bit input: B port enable/Read enable input
   REGCEBREGCE => REGCEBREGCE, -- 1-bit input: B port register enable/Register enable input
   RSTBRST => RSTBRST,         -- 1-bit input: B port set/reset input
   WEBWEU => WEBWEU,           -- 2-bit input: B port write enable input
   -- Port B Data: 16-bit (each) input: Port B data
   DIBDI => DIBDI,             -- 16-bit input: B port data/MSB data input
   DIPBDIP => DIPBDIP          -- 2-bit input: B port parity/MSB parity input
);

-- End of RAMB8BWER_inst instantiation
```

Verilog Instantiation Template

```
// RAMB8BWER: 8k-bit Data and 1k-bit Parity Configurable Synchronous Block RAM
//            Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

RAMB8BWER #(
   // DATA_WIDTH_A/DATA_WIDTH_B: 'If RAM_MODE="TDP": 0, 1, 2, 4, 9 or 18; If RAM_MODE="SDP": 36'
   .DATA_WIDTH_A(0),
   .DATA_WIDTH_B(0),
   // DOA_REG/DOB_REG: Optional output register (0 or 1)
   .DOA_REG(0),
   .DOB_REG(0),
   // EN_RSTRAM_A/EN_RSTRAM_B: Enable/disable RST
   .EN_RSTRAM_A("TRUE"),
   .EN_RSTRAM_B("TRUE"),
   // INITP_00 to INITP_03: Initial memory contents.
   .INITP_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
   // INIT_00 to INIT_1F: Initial memory contents.
   .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),
```

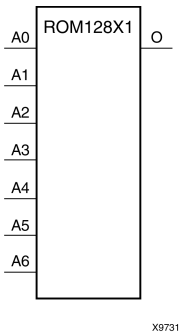
```
   .INIT_10(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_11(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_12(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_13(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .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_A/INIT_B: Initial values on output port
   .INIT_A(18'h00000),
   .INIT_B(18'h00000),
   // INIT_FILE: Not Supported
   .INIT_FILE("NONE"),                                                    // Do not modify
   // RAM_MODE: "SDP" or "TDP"
   .RAM_MODE("TDP"),
   // RSTTYPE: "SYNC" or "ASYNC"
   .RSTTYPE("SYNC"),
   // RST_PRIORITY_A/RST_PRIORITY_B: "CE" or "SR"
   .RST_PRIORITY_A("CE"),
   .RST_PRIORITY_B("CE"),
   // SIM_COLLISION_CHECK: Collision check enable "ALL", "WARNING_ONLY", "GENERATE_X_ONLY" or "NONE"
   .SIM_COLLISION_CHECK("ALL"),
   // SRVAL_A/SRVAL_B: Set/Reset value for RAM output
   .SRVAL_A(18'h00000),
   .SRVAL_B(18'h00000),
   // WRITE_MODE_A/WRITE_MODE_B: "WRITE_FIRST", "READ_FIRST", or "NO_CHANGE"
   .WRITE_MODE_A("WRITE_FIRST"),
   .WRITE_MODE_B("WRITE_FIRST")
)
RAMB8BWER_inst (
   // Port A Data: 16-bit (each) output: Port A data
   .DOADO(DOADO),             // 16-bit output: A port data/LSB data output
   .DOPADOP(DOPADOP),         // 2-bit output: A port parity/LSB parity output
   // Port B Data: 16-bit (each) output: Port B data
   .DOBDO(DOBDO),             // 16-bit output: B port data/MSB data output
   .DOPBDOP(DOPBDOP),         // 2-bit output: B port parity/MSB parity output
   // Port A Address/Control Signals: 13-bit (each) input: Port A address and control signals (write port
   // when RAM_MODE="SDP")
   .ADDRAWRADDR(ADDRAWRADDR), // 13-bit input: A port address/Write address input
   .CLKAWRCLK(CLKAWRCLK),     // 1-bit input: A port clock/Write clock input
   .ENAWREN(ENAWREN),         // 1-bit input: A port enable/Write enable input
   .REGCEA(REGCEA),           // 1-bit input: A port register enable input
   .RSTA(RSTA),               // 1-bit input: A port set/reset input
   .WEAWEL(WEAWEL),           // 2-bit input: A port write enable input
   // Port A Data: 16-bit (each) input: Port A data
   .DIADI(DIADI),             // 16-bit input: A port data/LSB data input
   .DIPADIP(DIPADIP),         // 2-bit input: A port parity/LSB parity input
   // Port B Address/Control Signals: 13-bit (each) input: Port B address and control signals (read port
   // when RAM_MODE="SDP")
   .ADDRBRDADDR(ADDRBRDADDR), // 13-bit input: B port address/Read address input
   .CLKBRDCLK(CLKBRDCLK),     // 1-bit input: B port clock/Read clock input
   .ENBRDEN(ENBRDEN),         // 1-bit input: B port enable/Read enable input
   .REGCEBREGCE(REGCEBREGCE), // 1-bit input: B port register enable/Register enable input
   .RSTBRST(RSTBRST),         // 1-bit input: B port set/reset input
   .WEBWEU(WEBWEU),           // 2-bit input: B port write enable input
   // Port B Data: 16-bit (each) input: Port B data
   .DIBDI(DIBDI),             // 16-bit input: B port data/MSB data input
   .DIPBDIP(DIPBDIP)          // 2-bit input: B port parity/MSB parity input
);

// End of RAMB8BWER_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA Block RAM User Guide*
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

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
CORE Generator™ and wizards	No
Macro support	No

Available Attributes

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

VHDL Instantiation Template

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

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


-- ROM128X1: 128 x 1 Asynchronous Distributed (LUT) ROM
--           Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of ROM128X1_inst instantiation
```

Verilog Instantiation Template

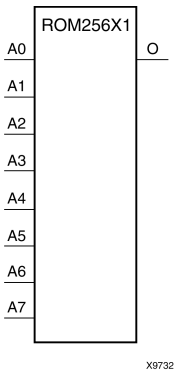
```
// ROM128X1: 128 x 1 Asynchronous Distributed (LUT) ROM
//           Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

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

// End of ROM128X1_inst instantiation
```

For More Information

See the *Spartan-6 FPGA User Documentation (User Guides and 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
CORE Generator™ and wizards	No
Macro support	No

Available Attributes

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

VHDL Instantiation Template

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

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


-- ROM256X1: 256 x 1 Asynchronous Distributed (LUT) ROM
--           Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of ROM256X1_inst instantiation
```

Verilog Instantiation Template

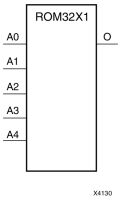
```
// ROM256X1: 256 x 1 Asynchronous Distributed (LUT) ROM
//           Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

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

// End of ROM256X1_inst instantiation
```

For More Information

See the *Spartan-6 FPGA User Documentation (User Guides and 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
CORE Generator™ and wizards	No
Macro support	No

Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Hexadecimal	Any 32-Bit Value	All zeros	Specifies the contents of the ROM.

VHDL Instantiation Template

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

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


-- ROM32X1: 32 x 1 Asynchronous Distributed (LUT) ROM
--          Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

ROM32X1_inst : ROM32X1
generic map (
   INIT => X"00000000")
port map (
   O => O,   -- ROM output
   A0 => A0, -- ROM address[0]
   A1 => A1, -- ROM address[1]
   A2 => A2, -- ROM address[2]
   A3 => A3, -- ROM address[3]
   A4 => A4  -- ROM address[4]
);
-- End of ROM32X1_inst instantiation
```

Verilog Instantiation Template

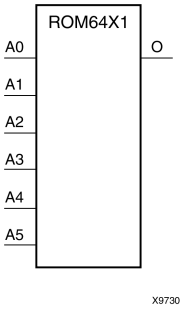
```
// ROM32X1: 32 x 1 Asynchronous Distributed (LUT) ROM
//          Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

ROM32X1 #(
   .INIT(32'h00000000) // Contents of ROM
) ROM32X1_inst (
   .O(O),   // ROM output
   .A0(A0), // ROM address[0]
   .A1(A1), // ROM address[1]
   .A2(A2), // ROM address[2]
   .A3(A3), // ROM address[3]
   .A4(A4)  // ROM address[4]
);

// End of ROM32X1_inst instantiation
```

For More Information

See the *Spartan-6 FPGA User Documentation (User Guides and 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
CORE Generator™ and wizards	No
Macro support	No

Available Attributes

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

VHDL Instantiation Template

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

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


-- ROM64X1: 64 x 1 Asynchronous Distributed (LUT) ROM
--          Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

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

-- End of ROM64X1_inst instantiation
```

Verilog Instantiation Template

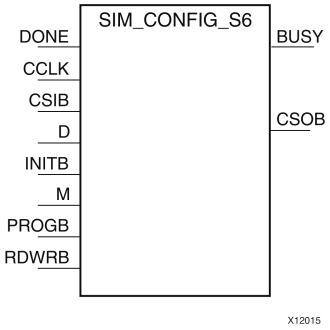
```
// ROM64X1: 64 x 1 Asynchronous Distributed (LUT) ROM
//          Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

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

// End of ROM64X1_inst instantiation
```

For More Information

See the *Spartan-6 FPGA User Documentation (User Guides and Data Sheets)*.

SIM_CONFIG_S6

Simulation: Configuration Simulation Model



Introduction

This simulation component allows the functional simulation of many of the common configuration interface, functions and commands to assist with board-level understanding and debug of configuration behaviors. The model can also simulate some startup-up behaviors such as the global set/reset (GSR) and global 3-state (GTS) assertion in the design. This model does not map to a specific primitive in the FPGA software and cannot be directly instantiated in the design, however it can be used in conjunction with the source design if specified either in a simulation-only file like a testbench or by some means guarded from synthesis so that it is not synthesized into the design netlist. This model may be used for either functional (RTL) simulation or timing simulation. This model is also indirectly used when instantiating the ICAP_SPARTAN6 in simulating configuration access to that component.

Port Descriptions

Port	Direction	Width	Function
BUSY	Output	1	This output pin is used during read back.
CSOB	Output	1	Parallel daisy-chain active-Low chip select output. Not used in single FPGA applications.
DONE	Inout	1	Active-High signal indicating configuration is complete: • 0 = FPGA not configured • 1 = FPGA configured
CCLK	Input	1	Configuration clock source for all configuration modes except JTAG.
CSIB	Input	1	Active-Low chip select to enable the SelectMAP data bus: • 0 = SelectMAP data bus enabled • 1 = SelectMAP data bus disabled
D	Input	32	Configuration and read back data bus, clocked on the rising edge of CCLK.
INITB	Input	1	Before the Mode pins are sampled, INIT_B is an input that can be held Low to delay configuration. After the Mode pins are sampled, INIT_B is an open-drain, active-Low output indicating whether a CRC error occurred during configuration: • 0 = CRC error • 1 = No CRC error When the SEU detection function is enabled, INIT_B is optionally driven Low when a read back CRC error is detected.

Port	Direction	Width	Function
M	Input	2	Mode pins - determine configuration mode.
PROGB	Input	1	Active-Low asynchronous full-chip reset.
RDWRB	Input	1	Determines the direction of the D[x:0] data bus: • 0 = inputs • 1 = outputs RDWR_B input can only be changed while CSI_B is deasserted, otherwise an ABORT occurs

Design Entry Method

Instantiation	In testbench or simulation-only file.
Inference	No
CORE Generator™ and wizards	No
Macro support	No

Xilinx suggests that you instantiate this in the testbench file and not an implementation file or file used during synthesis of the design. It may be used in conjunction with the design in order to help determine interaction and start-up sequences between configuration loading and device start-up. In general, a configuration bitstream file is to be used in conjunction with this model in order to observe configuration behavior.

More information on simulating and using this component can be found in the *Xilinx Synthesis and Simulation Design Guide*. Please refer to that guide for further detail on using this component.

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
DEVICE_ID	32-bit hexadecimal	Valid device ID codes	32'h00000000	Specify the Device ID code for the target device. Used during bitstream processing and device identification reads.

VHDL Instantiation Template

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

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


-- SIM_CONFIG_S6: Behavioral Simulation-only Model of FPGA SelectMap Configuration
--                Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

SIM_CONFIG_S6_inst : SIM_CONFIG_S6
generic map (
   DEVICE_ID => X"00000000")  -- Specifies the Pre-programmed Device ID value
port map (
   BUSY => BUSY,   -- 1-bit output Busy pin
   CSOB => CSOB,   -- 1-bit output chip select pin
   DONE => DONE,   -- 1-bit bi-directional Done pine
   CCLK => CCLK,   -- 1-bit input configuration clock
   D => D,         -- 8-bit bi-directional configuration data
   INITB =>INITB, -- 1-bit bi-directional INIT status pin
   M => M,         -- 3-bit input Mode pins
   PROGB => PROGB, -- 1-bit input Program pin
   RDWRB => RDWRB  -- 1-bit input Read/Write pin
```

```
);

-- End of SIM_CONFIG_S6_inst instantiation
```

Verilog Instantiation Template

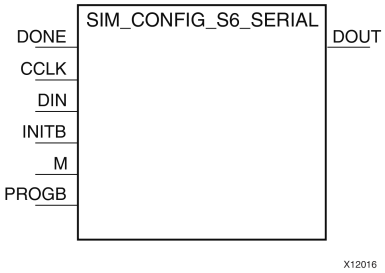
```
// SIM_CONFIG_S6: Behavioral Simulation-only Model of FPGA SelectMap Configuration
//                Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

SIM_CONFIG_S6 #(
   .DEVICE_ID(32'h00000000) // Specify DEVICE_ID
) SIM_CONFIG_S6_inst (
   .BUSY(BUSY), // 1-bit output Busy pin
   .CSOB(CSOB), // 1-bit output chip select pin
   .DONE(DONE), // 1-bit bi-directional Done pin
   .CCLK(CCLK), // 1-bit input configuration clock
   .CSIB(CSIB), // 1-bit input chip select
   .D(D),       // 16-bit bi-directional configuration data
   .INITB(INITB),  // 1-bit bi-directional INIT status pin
   .M(M),          // 2-bit input Mode pins
   .PROGB(PROGB),  // 1-bit input Program pin
   .RDWRB(RDWRB)   // 1-bit input Read/write pin
);

// End of SIM_CONFIG_S6_inst instantiation
```

For More Information

- See the *Synthesis and Simulation Design Guide*.
- See the *Spartan-6 FPGA Configuration User Guide*.

SIM_CONFIG_S6_SERIAL

Simulation: Serial Configuration Simulation Model



Introduction

This simulation component allows the functional simulation of many of the common serial configuration interface, functions and commands to assist with board-level understanding and debug of configuration behaviors. The model can also simulate some startup-up behaviors such as the global set/reset (GSR) and global 3-state (GTS) assertion in the design. This model does not map to a specific primitive in the FPGA software and cannot be directly instantiated in the design, however it can be used in conjunction with the source design if specified either in a simulation-only file like a testbench or by some means guarded from synthesis so that it is not synthesized into the design netlist. This model may be used for either functional (RTL) simulation or timing simulation.

Port Descriptions

Port	Direction	Width	Function
DONE	Inout	1	Active-High signal indicating configuration is complete: • 0 = FPGA not configured • 1 = FPGA configured
DOUT	Output	1	Serial data output for downstream daisy-chained devices. Data provided on the falling edge of CCLK.
CCLK	Input	1	Configuration clock source for all configuration modes except JTAG.
DIN	Input	1	Serial configuration data input, synchronous to rising CCLK edge.
INITB	Input	1	Before the Mode pins are sampled, INIT_B is an input that can be held Low to delay configuration. After the Mode pins are sampled, INIT_B is an open-drain, active-Low output indicating whether a CRC error occurred during configuration: • 0 = CRC error • 1 = No CRC error When the SEU detection function is enabled, INIT_B is optionally driven Low when a read back CRC error is detected.
M	Input	2	Mode pins - determine configuration mode.
PROGB	Input	1	Active-Low asynchronous full-chip reset.

Design Entry Method

Instantiation	In testbench or simulation-only file.
Inference	No
CORE Generator™ and wizards	No
Macro support	No

Xilinx suggests that you instantiate this in the testbench file and not an implementation file or file used during synthesis of the design. It may be used in conjunction with the design in order to help determine interaction and start-up sequences between configuration loading and device start-up. In general, a configuration bitstream file is to be used in conjunction with this model in order to observe configuration behavior.

More information on simulating and using this component can be found in the *Xilinx Synthesis and Simulation Design Guide*. Please refer to that guide for further detail on using this component.

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
DEVICE_ID	32-bit hexadecimal	Valid device ID codes	32'h00000000	Specify the Device ID code for the target device. Used during bitstream processing and device identification reads.

VHDL Instantiation Template

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

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


-- SIM_CONFIG_S6_SERIAL: Behavioral Simulation-only Model of FPGA Serial Configuration
--                       Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

SIM_CONFIG_S6_SERIAL_inst : SIM_CONFIG_S6_SERIAL
generic map (
   DEVICE_ID => X"00000000")  -- Specifies the Pre-programmed Device ID value
port map (
   DONE => DONE,    -- 1-bit bi-directional Done pine
   CCLK => CCLK,    -- 1-bit input configuration clock
   DIN => DIN,      -- 1-bit input configuration data
   INITB =>INITB,   -- 1-bit bi-directional INIT status pin
   M => M,          -- 3-bit input Mode pins
   PROGB => PROGB   -- 1-bit input Program pin
);

-- End of SIM_CONFIG_S6_SERIAL_inst instantiation
```

Verilog Instantiation Template

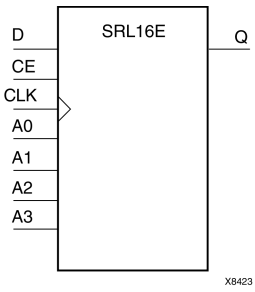
```
// SIM_CONFIG_S6_SERIAL: Behavioral Simulation-only Model of FPGA Serial Configuration
//                       Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

SIM_CONFIG_S6_SERIAL #(
   .DEVICE_ID(32'h00000000) // Specify DEVICE_ID
) SIM_CONFIG_S6_SERIAL_inst (
   .DONE(DONE),     // 1-bit bi-directional Done pin
   .CCLK(CCLK),     // 1-bit input configuration clock
   .DIN(DIN),       // 1-bit input configuration data
   .INITB(INITB),   // 1-bit bi-directional INIT status pin
   .M(M),           // 2-bit input Mode pins
   .PROGB(PROGB)    // 1-bit input Program pin
);

// End of SIM_CONFIG_S6_SERIAL_inst instantiation
```

For More Information

- See the *Synthesis and Simulation Design Guide*.
- See the *Spartan-6 FPGA Configuration User Guide*.

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				

Port Descriptions

Port	Direction	Width	Function
Q	Output	1	Shift register data output
D	Input	1	Shift register data input
CLK	Input	1	Clock
CE	Input	1	Active high clock enable
A	Input	4	Dynamic depth selection of the SRL • A=0000 ==> 1-bit shift length • A=1111 ==> 16-bit shift length

Design Entry Method

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

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
INIT	Hexa-decimal	Any 16-Bit Value	All zeros	Sets the initial value of content and output of shift register after configuration.

VHDL Instantiation Template

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

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


-- SRL16E: 16-bit shift register LUT with clock enable operating on posedge of clock
--        Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

SRL16E_inst : SRL16E
generic map (
   INIT => X"0000")
port map (
   Q => Q,       -- SRL data output
   A0 => A0,     -- Select[0] input
   A1 => A1,     -- Select[1] input
   A2 => A2,     -- Select[2] input
   A3 => A3,     -- Select[3] input
   CE => CE,     -- Clock enable input
   CLK => CLK,   -- Clock input
   D => D        -- SRL data input
);

-- End of SRL16E_inst instantiation
```

Verilog Instantiation Template

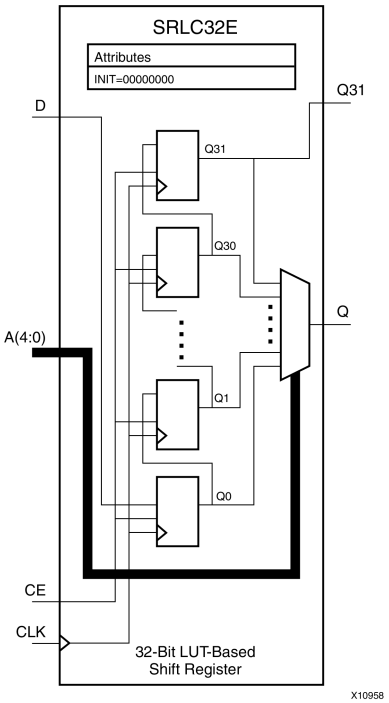
```
// SRL16E: 16-bit shift register LUT with clock enable operating on posedge of clock
//         Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

SRL16E #(
   .INIT(16'h0000) // Initial Value of Shift Register
) SRL16E_inst (
   .Q(Q),       // SRL data output
   .A0(A0),     // Select[0] input
   .A1(A1),     // Select[1] input
   .A2(A2),     // Select[2] input
   .A3(A3),     // Select[3] input
   .CE(CE),     // Clock enable input
   .CLK(CLK),   // Clock input
   .D(D)        // SRL data input
);

// End of SRL16E_inst instantiation
```

For More Information

See the *Spartan-6 FPGA User Documentation (User Guides and Data Sheets)*.

SRLC32E

Primitive: 32 Clock Cycle, Variable Length Shift Register Look-Up Table (LUT) with Clock Enable



Introduction

This design element is a variable length, 1 to 32 clock cycle shift register implemented within a single look-up table (LUT). The shift register can be of a fixed length, static length, or it can be dynamically adjusted by changing the address lines to the component. This element also features an active, high-clock enable and a cascading feature in which multiple SRLC32Es can be cascaded in order to create greater shift lengths.

Port Descriptions

Port	Direction	Width	Function
Q	Output	1	Shift register data output
Q31	Output	1	Shift register cascaded output (connect to the D input of a subsequent SRLC32E)
D	Input	1	Shift register data input
CLK	Input	1	Clock
CE	Input	1	Active high clock enable
A	Input	5	Dynamic depth selection of the SRL A=00000 ==> 1-bit shift length A=11111 ==> 32-bit shift length

Design Entry Method

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

If instantiated, the following connections should be made to this component:

- Connect the CLK input to the desired clock source, the D input to the data source to be shifted/stored and the Q output to either an FDCPE or an FDRSE input or other appropriate data destination.
- The CE clock enable pin can be connected to a clock enable signal in the design or else tied to a logic one if not used.
- The 5-bit A bus can either be tied to a static value between 0 and 31 to signify a fixed 1 to 32 bit static shift length, or else it can be tied to the appropriate logic to enable a varying shift depth anywhere between 1 and 32 bits.
- If you want to create a longer shift length than 32, connect the Q31 output pin to the D input pin of a subsequent SRLC32E to cascade and create larger shift registers.
- It is not valid to connect the Q31 output to anything other than another SRLC32E.
- The selectable Q output is still available in the cascaded mode, if needed.
- An optional INIT attribute consisting of a 32-bit Hexadecimal value can be specified to indicate the initial shift pattern of the shift register.
- (INIT[0] will be the first value shifted out.)

Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Hexa-decimal	Any 32-Bit Value	All zeros	Specifies the initial shift pattern of the SRLC32E.

VHDL Instantiation Template

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

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


-- SRLC32E: 32-bit variable length shift register LUT
--          with clock enable
--          Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

SRLC32E_inst : SRLC32E
generic map (
   INIT => X"00000000")
port map (
   Q => Q,         -- SRL data output
   Q31 => Q31,     -- SRL cascade output pin
   A => A,         -- 5-bit shift depth select input
   CE => CE,       -- Clock enable input
   CLK => CLK,     -- Clock input
   D => D          -- SRL data input
);

-- End of SRLC32E_inst instantiation
```

Verilog Instantiation Template

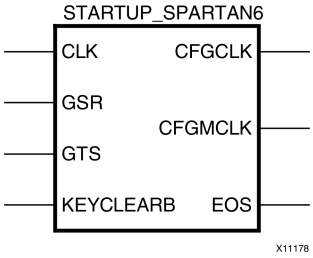
```
// SRLC32E: 32-bit variable length cascadable shift register LUT
//          with clock enable
//          Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

SRLC32E #(
   .INIT(32'h00000000) // Initial Value of Shift Register
) SRLC32E_inst (
   .Q(Q),     // SRL data output
   .Q31(Q31), // SRL cascade output pin
   .A(A),     // 5-bit shift depth select input
   .CE(CE),   // Clock enable input
   .CLK(CLK), // Clock input
   .D(D)      // SRL data input
);

// End of SRLC32E_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA Configurable Logic Block User Guide*.
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

STARTUP_SPARTAN6

Primitive: Spartan®-6 Global Set/Reset, Global 3-State and Configuration Start-Up Clock Interface



Introduction

This design element is used to interface device pins and logic to the Global Set/Reset (GSR) signal, the Global Tristate (GTS) dedicated routing, the internal configuration signals, or the input pins for the SPI PROM if an SPI PROM is used to configure the device. This primitive can also be used to specify a different clock for the device startup sequence at the end of configuring the device, and to access the configuration clock to the internal logic.

Port Descriptions

Port	Direction	Width	Function
CFGCLK	Output	1	Configuration logic main clock output.
CFGMCLK	Output	1	Configuration internal oscillator clock output.
CLK	Input	1	User startup-clock input
EOS	Output	1	Active high output signal indicates the End Of Configuration.
GSR	Input	1	Global Set/Reset (GSR) input (GSR cannot be used for the port name).
GTS	Input	1	Global Tristate (GTS) input (GTS cannot be used for the port name).
KEYCLEARB	Input	1	Clear AES Decrypter Key input from Battery-Backed RAM (BBRAM).

Design Entry Method

Instantiation	Recommended
Inference	No
CORE Generator™ 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 does not 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. 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_SPARTAN6: STARTUP Block
--                   Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

STARTUP_SPARTAN6_inst : STARTUP_SPARTAN6
port map (
   CFGCLK => CFGCLK,       -- 1-bit output: Configuration logic main clock output.
   CFGMCLK => CFGMCLK,     -- 1-bit output: Configuration internal oscillator clock output.
   EOS => EOS,             -- 1-bit output: Active high output signal indicates the End Of Configuration.
   CLK => CLK,             -- 1-bit input: User startup-clock input
   GSR => GSR,             -- 1-bit input: Global Set/Reset input (GSR cannot be used for the port name)
   GTS => GTS,             -- 1-bit input: Global 3-state input (GTS cannot be used for the port name)
   KEYCLEARB => KEYCLEARB  -- 1-bit input: Clear AES Decrypter Key input from Battery-Backed RAM (BBRAM)
);

-- End of STARTUP_SPARTAN6_inst instantiation
```

Verilog Instantiation Template

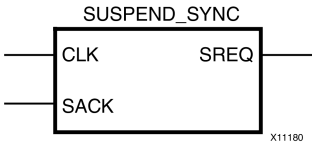
```
// STARTUP_SPARTAN6: STARTUP Block
//                   Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

STARTUP_SPARTAN6 STARTUP_SPARTAN6_inst (
   .CFGCLK(CFGCLK),       // 1-bit output: Configuration logic main clock output.
   .CFGMCLK(CFGMCLK),     // 1-bit output: Configuration internal oscillator clock output.
   .EOS(EOS),             // 1-bit output: Active high output signal indicates the End Of Configuration.
   .CLK(CLK),             // 1-bit input: User startup-clock input
   .GSR(GSR),             // 1-bit input: Global Set/Reset input (GSR cannot be used for the port name)
   .GTS(GTS),             // 1-bit input: Global 3-state input (GTS cannot be used for the port name)
   .KEYCLEARB(KEYCLEARB)  // 1-bit input: Clear AES Decrypter Key input from Battery-Backed RAM (BBRAM)
);

// End of STARTUP_SPARTAN6_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA Configuration User Guide*
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

SUSPEND_SYNC

Primitive: Suspend Mode Access



Introduction

The SUSPEND primitive extends the capabilities of the user to synchronize the design for applications using the suspend mode. It uses a three-pin interface to allow synchronization of the trigger to start the suspend mode, even when there are several clock domains requiring synchronization.

Port Descriptions

Port	Direction	Width	Function
CLK	Input	1	User clock input.
SACK	Input	1	SUSPEND acknowledgement output.
SREQ	Output	1	Suspend request output.

Design Entry Method

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

VHDL Instantiation Template

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

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


-- SUSPEND_SYNC: Suspend Mode Access
--               Spartan-6
-- Xilinx HDL Libraries Guide, version 14.1

SUSPEND_SYNC_inst : SUSPEND_SYNC
port map (
   SREQ => SREQ, -- 1-bit output: Suspend request output
   CLK => CLK,   -- 1-bit input: User clock input
   SACK => SACK  -- 1-bit input: SUSPEND acknowledgement output
);

-- End of SUSPEND_SYNC_inst instantiation
```

Verilog Instantiation Template

```
// SUSPEND_SYNC: Suspend Mode Access
//               Spartan-6
// Xilinx HDL Libraries Guide, version 14.1

SUSPEND_SYNC SUSPEND_SYNC_inst (
   .SREQ(SREQ), // 1-bit output: Suspend request output
   .CLK(CLK),   // 1-bit input: User clock input
   .SACK(SACK)  // 1-bit input: SUSPEND acknowledgement output
);

// End of SUSPEND_SYNC_inst instantiation
```

For More Information

- See the *Spartan-6 FPGA Configuration User Guide*
- See the *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.