Virtex-6 Libraries Guide for HDL Designs

UG623 (v 14.2) July 25, 2012

XILINX®

Notice of Disclaimer

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 Virtex®-6 devices, 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

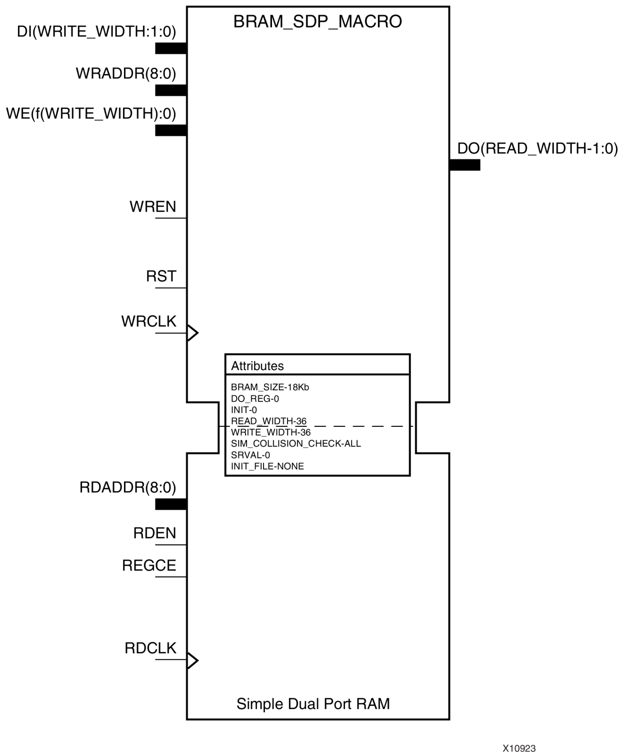
This section describes the unimacros that can be used with Virtex®-6 devices. The unimacros are organized alphabetically.

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 36Kb or 18Kb 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
72 - 37	36Kb	9	8
36 - 19	36Kb	10	4
	18Kb	9	
18 - 10	36Kb	11	2
	18Kb	10	
9 - 5	36Kb	12	1
	18Kb	11	
4 - 3	36Kb	13	1
	18Kb	12	
2	36Kb	14	1
	18Kb	13	
1	36Kb	15	1
	18Kb	14	

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	"36Kb", "18Kb"	"18Kb"	Configures RAM as "36Kb" or "18Kb" 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	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.

Attribute	Data Type	Allowed Values	Default	Description
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.
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
--                 Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

-- 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  --
-- ============|===========|=============|==============|============--
--    37-72    |  "36Kb"   |      512    |     9-bit    |    8-bit   --
--    19-36    |  "36Kb"   |     1024    |    10-bit    |    4-bit   --
--    19-36    |  "18Kb"   |      512    |     9-bit    |    4-bit   --
--    10-18    |  "36Kb"   |     2048    |    11-bit    |    2-bit   --
--    10-18    |  "18Kb"   |     1024    |    10-bit    |    2-bit   --
--     5-9     |  "36Kb"   |     4096    |    12-bit    |    1-bit   --
--     5-9     |  "18Kb"   |     2048    |    11-bit    |    1-bit   --
--     3-4     |  "36Kb"   |     8192    |    13-bit    |    1-bit   --
--     3-4     |  "18Kb"   |     4096    |    12-bit    |    1-bit   --
--       2     |  "36Kb"   |    16384    |    14-bit    |    1-bit   --
--       2     |  "18Kb"   |     8192    |    13-bit    |    1-bit   --
--       1     |  "36Kb"   |    32768    |    15-bit    |    1-bit   --
--       1     |  "18Kb"   |    16384    |    14-bit    |    1-bit   --
-----------------------------------------------------------------------

BRAM_SDP_MACRO_inst : BRAM_SDP_MACRO
generic map (
   BRAM_SIZE => "18Kb", -- Target BRAM, "18Kb" or "36Kb"
   DEVICE => "VIRTEX6", -- Target device: "VIRTEX5", "VIRTEX6", "SPARTAN6"
   WRITE_WIDTH => 0,    -- Valid values are 1-72 (37-72 only valid when BRAM_SIZE="36Kb")
   READ_WIDTH => 0,     -- Valid values are 1-72 (37-72 only valid when BRAM_SIZE="36Kb")
   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"000000000000000000", --  Set/Reset value for port output
   INIT => X"000000000000000000", --  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",
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 INIT_xx are valid when configured as 36Kb
INIT_40 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_41 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_42 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_43 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_44 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_45 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_46 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_47 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_48 => X"0000000000000000000000000000000000000000000000000000000000000000",
```

```
   INIT_49 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_4A => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_4B => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_4C => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_4D => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_4E => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_4F => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_50 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_51 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_52 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_53 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_54 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_55 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_56 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_57 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_58 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_59 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_5A => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_5B => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_5C => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_5D => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_5E => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_5F => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_60 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_61 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_62 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_63 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_64 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_65 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_66 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_67 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_68 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_69 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_6A => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_6B => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_6C => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_6D => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_6E => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_6F => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_70 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_71 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_72 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_73 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_74 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_75 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_76 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_77 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_78 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_79 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_7A => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_7B => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_7C => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_7D => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_7E => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_7F => X"0000000000000000000000000000000000000000000000000000000000000000",

   -- The next set of INITP_xx are for the parity bits
   INITP_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_07 => X"0000000000000000000000000000000000000000000000000000000000000000",

   -- The next set of INIT_xx are valid when configured as 36Kb
   INITP_08 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_09 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_0A => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_0B => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_0C => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_0D => X"0000000000000000000000000000000000000000000000000000000000000000",
```

```
   INITP_0E => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_0F => 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
//                 Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

///////////////////////////////////////////////////////////////////////
//  READ_WIDTH | BRAM_SIZE | READ Depth  | RDADDR Width |            //
// WRITE_WIDTH |           | WRITE Depth | WRADDR Width |  WE Width  //
// ============|===========|=============|==============|============//
//    37-72    |  "36Kb"   |     512     |     9-bit    |    8-bit   //
//    19-36    |  "36Kb"   |    1024     |    10-bit    |    4-bit   //
//    19-36    |  "18Kb"   |     512     |     9-bit    |    4-bit   //
//    10-18    |  "36Kb"   |    2048     |    11-bit    |    2-bit   //
//    10-18    |  "18Kb"   |    1024     |    10-bit    |    2-bit   //
//     5-9     |  "36Kb"   |    4096     |    12-bit    |    1-bit   //
//     5-9     |  "18Kb"   |    2048     |    11-bit    |    1-bit   //
//     3-4     |  "36Kb"   |    8192     |    13-bit    |    1-bit   //
//     3-4     |  "18Kb"   |    4096     |    12-bit    |    1-bit   //
//       2     |  "36Kb"   |   16384     |    14-bit    |    1-bit   //
//       2     |  "18Kb"   |    8192     |    13-bit    |    1-bit   //
//       1     |  "36Kb"   |   32768     |    15-bit    |    1-bit   //
//       1     |  "18Kb"   |   16384     |    14-bit    |    1-bit   //
///////////////////////////////////////////////////////////////////////

BRAM_SDP_MACRO #(
   .BRAM_SIZE("18Kb"), // Target BRAM, "18Kb" or "36Kb"
   .DEVICE("VIRTEX6"), // Target device: "VIRTEX5", "VIRTEX6", "SPARTAN6"
   .WRITE_WIDTH(0),    // Valid values are 1-72 (37-72 only valid when BRAM_SIZE="36Kb")
   .READ_WIDTH(0),     // Valid values are 1-72 (37-72 only valid when BRAM_SIZE="36Kb")
   .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 forr port output
   .INIT(72'h000000000000000000),  // Initial values on output port
   .WRITE_MODE("WRITE_FIRST"),  // Specify "READ_FIRST" forr same clock or synchronous clocks
                                //   Specify "WRITE_FIRST for asynchronous clocks on ports
   .INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_05(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_06(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_07(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_08(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_09(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_0A(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_0B(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_0C(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_0D(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_0E(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_0F(256'h0000000000000000000000000000000000000000000000000000000000000000),
```

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

// The next set of INIT_xx are valid when configured as 36Kb
.INIT_40(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_41(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_42(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_43(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_44(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_45(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_46(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_47(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_48(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_49(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_4A(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_4B(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_4C(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_4D(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_4E(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_4F(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_50(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_51(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_52(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_53(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_54(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_55(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_56(256'h0000000000000000000000000000000000000000000000000000000000000000),
```

```
   .INIT_57(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_58(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_59(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_5A(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_5B(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_5C(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_5D(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_5E(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_5F(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_60(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_61(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_62(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_63(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_64(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_65(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_66(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_67(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_68(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_69(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_6A(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_6B(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_6C(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_6D(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_6E(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_6F(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_70(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_71(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_72(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_73(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_74(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_75(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_76(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_77(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_78(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_79(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_7A(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_7B(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_7C(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_7D(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_7E(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_7F(256'h0000000000000000000000000000000000000000000000000000000000000000),

   // The next set of INITP_xx are forr the parity bits
   .INITP_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_05(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_06(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_07(256'h0000000000000000000000000000000000000000000000000000000000000000),

   // The next set of INITP_xx are valid when configured as 36Kb
   .INITP_08(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_09(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_0A(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_0B(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_0C(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_0D(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_0E(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_0F(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
```

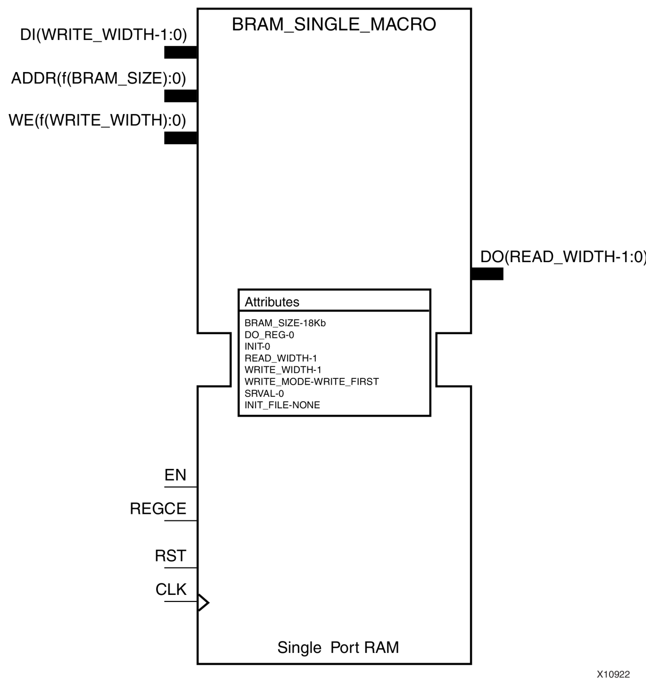
```
);
// End of BRAM_SDP_MACRO_inst instantiation
```

For More Information

See the *Virtex-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 36Kb or 18Kb 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	READ_WIDTH	BRAM_SIZE	ADDR	WE
37 - 72	37 - 72	36Kb	9	8
	36 - 19		10	
	18 - 10		11	
	9 - 5		12	
	4 - 3		13	
	2		14	
	1		15	
36 - 19	36 - 19	36Kb	10	4
	18-10		11	
	9 - 5		12	
	4 - 3		13	
	2		14	
	1		15	
18 - 10	36 - 19	36Kb	11	2
	18-10		11	
	9 - 5		12	
	4 - 3		13	
	2		14	
	1		15	
9 - 5	36-19	36Kb	12	1
	18-10		12	
	9 - 5		12	
	4 - 3		13	
	2		14	
	1		15	
4 - 3	36-19	36Kb	13	1
	18-10		13	
	9 - 5		13	
	4 - 3		13	
	2		14	
	1		15	
2	36-19	36Kb	14	1
	18-10		14	
	9 - 5		14	
	4 - 3		14	
	2		14	
	1		15	

WRITE_WIDTH	READ_WIDTH	BRAM_SIZE	ADDR	WE
1	36 - 19	36Kb	15	1
	18 - 10		15	
	9 - 5		15	
	3 - 4		15	
	2		15	
	1		15	
18-10	18-10	18Kb	10	2
	9 - 5		11	
	4 - 3		12	
	2		13	
	1		14	
9 - 5	18-10	18Kb	11	1
	9 - 5		11	
	4 - 3		12	
	2		13	
	1		14	
4 - 3	18-10	18Kb	12	1
	9 - 5		12	
	4 - 3		12	
	2		13	
	1		14	
2	18-10	18Kb	13	1
	9 - 5		13	
	4 - 3		13	
	2		13	
	1		14	
1	18-10	18Kb	14	1
	9 - 5		14	
	4 - 3		14	
	2		14	
	1		14	

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	"36Kb", "18Kb"	"18Kb"	Configures RAM as 36Kb or 18Kb 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. 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.
WRITE_MODE	String	"READ_FIRST", "WRITE_FIRST", "NO_CHANGE"	"WRITE_FIRST"	Specifies write mode to the memory
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
--                    Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

-- 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  --
-- ============|===========|=============|============|============--
--    37-72    |  "36Kb"   |     512     |    9-bit   |    8-bit   --
--    19-36    |  "36Kb"   |    1024     |   10-bit   |    4-bit   --
--    19-36    |  "18Kb"   |     512     |    9-bit   |    4-bit   --
--    10-18    |  "36Kb"   |    2048     |   11-bit   |    2-bit   --
--    10-18    |  "18Kb"   |    1024     |   10-bit   |    2-bit   --
--     5-9     |  "36Kb"   |    4096     |   12-bit   |    1-bit   --
--     5-9     |  "18Kb"   |    2048     |   11-bit   |    1-bit   --
--     3-4     |  "36Kb"   |    8192     |   13-bit   |    1-bit   --
--     3-4     |  "18Kb"   |    4096     |   12-bit   |    1-bit   --
--       2     |  "36Kb"   |   16384     |   14-bit   |    1-bit   --
--       2     |  "18Kb"   |    8192     |   13-bit   |    1-bit   --
--       1     |  "36Kb"   |   32768     |   15-bit   |    1-bit   --
--       1     |  "18Kb"   |   16384     |   14-bit   |    1-bit   --
---------------------------------------------------------------------


BRAM_SINGLE_MACRO_inst : BRAM_SINGLE_MACRO
generic map (
   BRAM_SIZE => "18Kb", -- Target BRAM, "18Kb" or "36Kb"
   DEVICE => "VIRTEX6", -- 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 1-72 (37-72 only valid when BRAM_SIZE="36Kb")
   READ_WIDTH => 0,   -- Valid values are 1-72 (37-72 only valid when BRAM_SIZE="36Kb")
   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",
    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 INIT_xx are valid when configured as 36Kb
    INIT_40 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_41 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_42 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_43 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_44 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_45 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_46 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_47 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_48 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_49 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_4A => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_4B => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_4C => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_4D => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_4E => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_4F => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_50 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_51 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_52 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_53 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_54 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_55 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_56 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_57 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_58 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_59 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_5A => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_5B => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_5C => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_5D => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_5E => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_5F => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_60 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_61 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_62 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_63 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_64 => X"0000000000000000000000000000000000000000000000000000000000000000",
```

```
   INIT_65 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_66 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_67 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_68 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_69 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_6A => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_6B => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_6C => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_6D => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_6E => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_6F => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_70 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_71 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_72 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_73 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_74 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_75 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_76 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_77 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_78 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_79 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_7A => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_7B => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_7C => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_7D => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_7E => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_7F => X"0000000000000000000000000000000000000000000000000000000000000000",

   -- The next set of INITP_xx are for the parity bits
   INITP_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_07 => X"0000000000000000000000000000000000000000000000000000000000000000",

   -- The next set of INIT_xx are valid when configured as 36Kb
   INITP_08 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_09 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_0A => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_0B => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_0C => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_0D => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_0E => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_0F => 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
//                    Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

/////////////////////////////////////////////////////////////////////
//  READ_WIDTH | BRAM_SIZE | READ Depth  | ADDR Width |            //
// WRITE_WIDTH |           | WRITE Depth |            |  WE Width  //
// ============|===========|=============|============|============//
//    37-72    |  "36Kb"   |      512    |     9-bit  |    8-bit   //
```

```
//     19-36     |  "36Kb"    |     1024     |   10-bit    |     4-bit   //
//     19-36     |  "18Kb"    |      512     |    9-bit    |     4-bit   //
//     10-18     |  "36Kb"    |     2048     |   11-bit    |     2-bit   //
//     10-18     |  "18Kb"    |     1024     |   10-bit    |     2-bit   //
//      5-9      |  "36Kb"    |     4096     |   12-bit    |     1-bit   //
//      5-9      |  "18Kb"    |     2048     |   11-bit    |     1-bit   //
//      3-4      |  "36Kb"    |     8192     |   13-bit    |     1-bit   //
//      3-4      |  "18Kb"    |     4096     |   12-bit    |     1-bit   //
//        2      |  "36Kb"    |    16384     |   14-bit    |     1-bit   //
//        2      |  "18Kb"    |     8192     |   13-bit    |     1-bit   //
//        1      |  "36Kb"    |    32768     |   15-bit    |     1-bit   //
//        1      |  "18Kb"    |    16384     |   14-bit    |     1-bit   //
//////////////////////////////////////////////////////////////////////

BRAM_SINGLE_MACRO #(
   .BRAM_SIZE("18Kb"), // Target BRAM, "18Kb" or "36Kb"
   .DEVICE("VIRTEX6"), // 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 1-72 (37-72 only valid when BRAM_SIZE="36Kb")
   .READ_WIDTH(0),  // Valid values are 1-72 (37-72 only valid when BRAM_SIZE="36Kb")
   .SRVAL(36'h000000000), // Set/Reset value forr 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),
   .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 INIT_xx are valid when configured as 36Kb
.INIT_40(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_41(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_42(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_43(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_44(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_45(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_46(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_47(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_48(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_49(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_4A(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_4B(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_4C(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_4D(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_4E(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_4F(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_50(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_51(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_52(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_53(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_54(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_55(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_56(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_57(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_58(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_59(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_5A(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_5B(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_5C(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_5D(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_5E(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_5F(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_60(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_61(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_62(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_63(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_64(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_65(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_66(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_67(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_68(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_69(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_6A(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_6B(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_6C(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_6D(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_6E(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_6F(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_70(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_71(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_72(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_73(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_74(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_75(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_76(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_77(256'h0000000000000000000000000000000000000000000000000000000000000000),
```

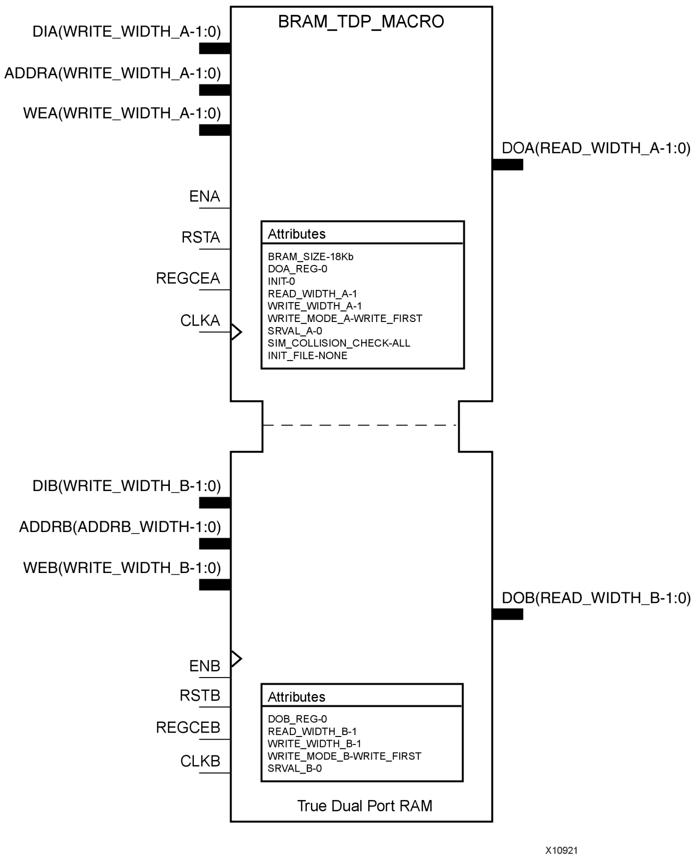
```
   .INIT_78(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_79(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_7A(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_7B(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_7C(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_7D(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_7E(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_7F(256'h0000000000000000000000000000000000000000000000000000000000000000),

   // The next set of INITP_xx are forr the parity bits
   .INITP_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_05(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_06(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_07(256'h0000000000000000000000000000000000000000000000000000000000000000),

   // The next set of INIT_xx are valid when configured as 36Kb
   .INITP_08(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_09(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_0A(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_0B(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_0C(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_0D(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_0E(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_0F(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 *Virtex-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 36kb or 18kb 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	READ_WIDTH_A/B-DOA/DOB	BRAM_SIZE	ADDRA/B	WEA/B
36 - 19	36 - 19	36Kb	10	4
	18-10		11	
	9 - 5		12	
	4 - 3		13	
	2		14	
	1		15	
18 - 10	36 - 19	36Kb	11	2
	18-10		11	
	9 - 5		12	
	4 - 3		13	
	2		14	
	1		15	
9 - 5	36-19	36Kb	12	1
	18-10		12	
	9 - 5		12	
	4 - 3		13	
	2		14	
	1		15	
4 - 3	36-19	36Kb	13	1
	18-10		13	
	9 - 5		13	
	4 - 3		13	
	2		14	
	1		15	

WRITE_WIDTH_A/B-DIA/DIB	READ_WIDTH_A/B-DOA/DOB	BRAM_SIZE	ADDRA/B	WEA/B
2	36-19	36Kb	14	1
	18-10		14	
	9 - 5		14	
	4 - 3		14	
	2		14	
	1		15	
1	36-19	36Kb	15	1
	18-10		15	
	9 - 5		15	
	4 - 3		15	
	2		15	
	1		15	
18-10	18-10	18Kb	10	2
	9 - 5		11	
	4 - 3		12	
	2		13	
	1		14	
9 - 5	18-10	18Kb	11	1
	9 - 5		11	
	4 - 3		12	
	2		13	
	1		14	
4 - 3	18-10	18Kb	12	1
	9 - 5		12	
	4 - 3		12	
	2		13	
	1		14	
2	18-10	18Kb	13	1
	9 - 5		13	
	4 - 3		13	
	2		13	
	1		14	
1	18-10	18Kb	14	1
	9 - 5		14	
	4 - 3		14	
	2		14	
	1		14	

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	"36Kb", "18Kb"	"18Kb"	Configures RAM as 36Kb or 18Kb 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. 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)
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.

Attribute(s)	Data Type	Allowed Values	Default	Description
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
--                 Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

-- 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      |  "36Kb"   |   1024    |    10-bit     |    4-bit    --
--     10-18      |  "36Kb"   |   2048    |    11-bit     |    2-bit    --
--     10-18      |  "18Kb"   |   1024    |    10-bit     |    2-bit    --
--      5-9       |  "36Kb"   |   4096    |    12-bit     |    1-bit    --
--      5-9       |  "18Kb"   |   2048    |    11-bit     |    1-bit    --
--      3-4       |  "36Kb"   |   8192    |    13-bit     |    1-bit    --
--      3-4       |  "18Kb"   |   4096    |    12-bit     |    1-bit    --
--        2       |  "36Kb"   |  16384    |    14-bit     |    1-bit    --
--        2       |  "18Kb"   |   8192    |    13-bit     |    1-bit    --
--        1       |  "36Kb"   |  32768    |    15-bit     |    1-bit    --
--        1       |  "18Kb"   |  16384    |    14-bit     |    1-bit    --
--------------------------------------------------------------------------


BRAM_TDP_MACRO_inst : BRAM_TDP_MACRO
generic map (
   BRAM_SIZE => "18Kb", -- Target BRAM, "18Kb" or "36Kb"
   DEVICE => "VIRTEX6", -- 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 (19-36 only valid when BRAM_SIZE="36Kb")
   READ_WIDTH_B => 0,   -- Valid values are 1-36 (19-36 only valid when BRAM_SIZE="36Kb")
   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 (19-36 only valid when BRAM_SIZE="36Kb")
   WRITE_WIDTH_B => 0, -- Valid values are 1-36 (19-36 only valid when BRAM_SIZE="36Kb")
   -- The following INIT_xx declarations specify the initial contents of the RAM
   INIT_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
```

```
INIT_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_07 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_08 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_09 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_0F => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_10 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_11 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_12 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_13 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_14 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_15 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_16 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_17 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_18 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_19 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1F => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_20 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_21 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_22 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_23 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_24 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_25 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_26 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_27 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_28 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_29 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2F => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_30 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_31 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_32 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_33 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_34 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_35 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_36 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_37 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_38 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_39 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3F => X"0000000000000000000000000000000000000000000000000000000000000000",

-- The next set of INIT_xx are valid when configured as 36Kb
INIT_40 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_41 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_42 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_43 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_44 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_45 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_46 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_47 => X"0000000000000000000000000000000000000000000000000000000000000000",
```

```
INIT_48 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_49 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_4A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_4B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_4C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_4D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_4E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_4F => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_50 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_51 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_52 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_53 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_54 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_55 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_56 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_57 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_58 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_59 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_5A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_5B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_5C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_5D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_5E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_5F => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_60 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_61 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_62 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_63 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_64 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_65 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_66 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_67 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_68 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_69 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_6A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_6B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_6C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_6D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_6E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_6F => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_70 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_71 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_72 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_73 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_74 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_75 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_76 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_77 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_78 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_79 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_7A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_7B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_7C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_7D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_7E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_7F => X"0000000000000000000000000000000000000000000000000000000000000000",

-- The next set of INITP_xx are for the parity bits
INITP_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_07 => X"0000000000000000000000000000000000000000000000000000000000000000",

-- The next set of INIT_xx are valid when configured as 36Kb
INITP_08 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_09 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0A => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0B => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0C => X"0000000000000000000000000000000000000000000000000000000000000000",
```

```
   INITP_0D => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_0E => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_0F => X"0000000000000000000000000000000000000000000000000000000000000000")
 port map (
   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
```

Verilog Instantiation Template

```
// BRAM_TDP_MACRO: True Dual Port RAM
//                 Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

//////////////////////////////////////////////////////////////////////////
// DATA_WIDTH_A/B | BRAM_SIZE | RAM Depth | ADDRA/B Width | WEA/B Width //
// ===============|===========|===========|===============|=============//
//     19-36      |  "36Kb"   |    1024   |    10-bit     |    4-bit    //
//     10-18      |  "36Kb"   |    2048   |    11-bit     |    2-bit    //
//     10-18      |  "18Kb"   |    1024   |    10-bit     |    2-bit    //
//      5-9       |  "36Kb"   |    4096   |    12-bit     |    1-bit    //
//      5-9       |  "18Kb"   |    2048   |    11-bit     |    1-bit    //
//      3-4       |  "36Kb"   |    8192   |    13-bit     |    1-bit    //
//      3-4       |  "18Kb"   |    4096   |    12-bit     |    1-bit    //
//        2       |  "36Kb"   |   16384   |    14-bit     |    1-bit    //
//        2       |  "18Kb"   |    8192   |    13-bit     |    1-bit    //
//        1       |  "36Kb"   |   32768   |    15-bit     |    1-bit    //
//        1       |  "18Kb"   |   16384   |    14-bit     |    1-bit    //
//////////////////////////////////////////////////////////////////////////

BRAM_TDP_MACRO #(
   .BRAM_SIZE("18Kb"), // Target BRAM: "18Kb" or "36Kb"
   .DEVICE("VIRTEX6"), // 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 (19-36 only valid when BRAM_SIZE="36Kb")
   .READ_WIDTH_B (0),   // Valid values are 1-36 (19-36 only valid when BRAM_SIZE="36Kb")
   .SIM_COLLISION_CHECK ("ALL"), // Collision check enable "ALL", "WARNING_ONLY",
                                 //   "GENERATE_X_ONLY" or "NONE"
   .SRVAL_A(36'h00000000), // Set/Reset value forr port A output
   .SRVAL_B(36'h00000000), // Set/Reset value forr 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 (19-36 only valid when BRAM_SIZE="36Kb")
   .WRITE_WIDTH_B(0), // Valid values are 1-36 (19-36 only valid when BRAM_SIZE="36Kb")
   .INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_05(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_06(256'h0000000000000000000000000000000000000000000000000000000000000000),
```

```
.INIT_07(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_08(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_09(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0A(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0B(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0C(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0D(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0E(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_0F(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_10(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_11(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_12(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_13(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_14(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_15(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_16(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_17(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_18(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_19(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_1A(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_1B(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_1C(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_1D(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_1E(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_1F(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_20(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_21(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_22(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_23(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_24(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_25(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_26(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_27(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_28(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_29(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_2A(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_2B(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_2C(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_2D(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_2E(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_2F(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_30(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_31(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_32(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_33(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_34(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_35(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_36(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_37(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_38(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_39(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3A(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3B(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3C(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3D(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3E(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3F(256'h0000000000000000000000000000000000000000000000000000000000000000),

// The next set of INIT_xx are valid when configured as 36Kb
.INIT_40(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_41(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_42(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_43(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_44(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_45(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_46(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_47(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_48(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_49(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_4A(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_4B(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_4C(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_4D(256'h0000000000000000000000000000000000000000000000000000000000000000),
```

```
   .INIT_4E(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_4F(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_50(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_51(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_52(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_53(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_54(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_55(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_56(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_57(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_58(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_59(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_5A(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_5B(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_5C(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_5D(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_5E(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_5F(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_60(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_61(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_62(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_63(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_64(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_65(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_66(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_67(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_68(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_69(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_6A(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_6B(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_6C(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_6D(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_6E(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_6F(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_70(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_71(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_72(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_73(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_74(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_75(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_76(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_77(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_78(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_79(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_7A(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_7B(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_7C(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_7D(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_7E(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_7F(256'h0000000000000000000000000000000000000000000000000000000000000000),

   // The next set of INITP_xx are forr the parity bits
   .INIT_FF(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .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),

   // The next set of INITP_xx are valid when configured as 36Kb
   .INITP_08(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_09(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_0A(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_0B(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_0C(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_0D(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_0E(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_0F(256'h0000000000000000000000000000000000000000000000000000000000000000)
) BRAM_TDP_MACRO_inst (
   .DOA(DOA),       // Output port-A data, width defined by READ_WIDTH_A parameter
```

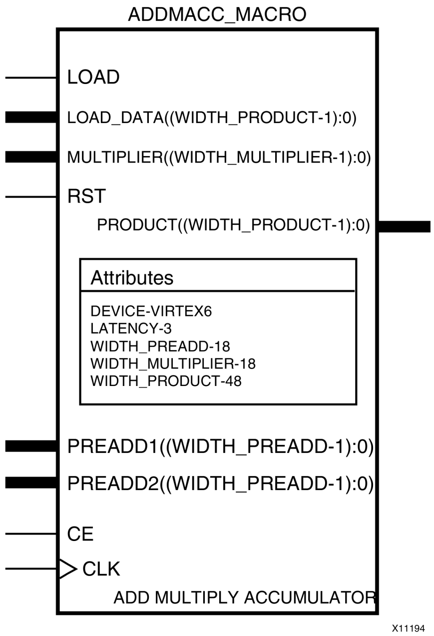
```
   .DOB(DOB),       // Output port-B data, width defined by READ_WIDTH_B parameter
   .ADDRA(ADDRA),   // Input port-A address, width defined by Port A depth
   .ADDRB(ADDRB),   // Input port-B address, width defined by Port B depth
   .CLKA(CLKA),     // 1-bit input port-A clock
   .CLKB(CLKB),     // 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 *Virtex-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

Name	Direction	Width	Function
LOAD_DATA	Input	Variable, see WIDTH_PRODUCT attribute.	In a DSP slice, when LOAD is asserted, loads P with A*B+LOAD_DATA.
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
--                Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

ADDMACC_MACRO_inst : ADDMACC_MACRO
generic map (
   DEVICE => "VIRTEX6",  -- 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

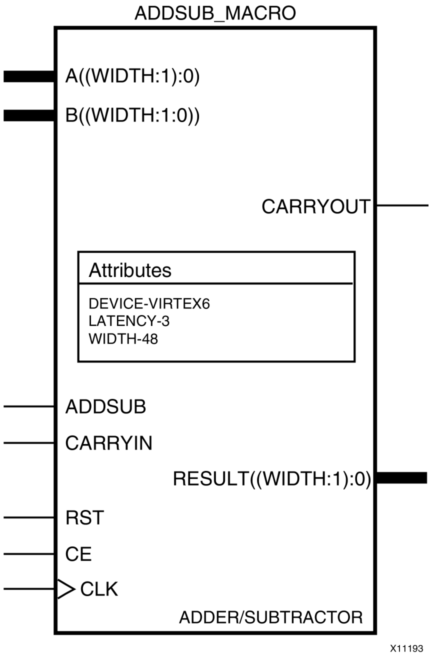
```
// ADDMACC_MACRO: Variable width & latency - Pre-Add -> Multiplier -> Accumulate
//                function implemented in a DSP48E
//                Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

ADDMACC_MACRO #(
   .DEVICE("VIRTEX6"),    // Target Device: "VIRTEX6", "SPARTAN6"
   .LATENCY(4),           // Desired clock cycle latency, 0-4
   .WIDTH_PREADD(25),     // Pre-adder input width, 1-25
   .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 *Virtex-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 / Subtractor implemented in a DSP48E
--                 Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

ADDSUB_MACRO_inst : ADDSUB_MACRO
generic map (
   DEVICE => "VIRTEX6", -- 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

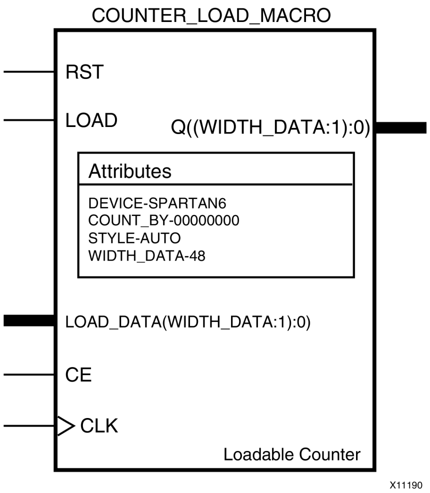
```
// ADDSUB_MACRO: Variable width & latency - Adder / Subtracter implemented in a DSP48E
//               Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

ADDSUB_MACRO #(
   .DEVICE("VIRTEX6"), // 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 *Virtex-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
--                     Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

COUNTER_LOAD_MACRO_inst : COUNTER_LOAD_MACRO
generic map (
   COUNT_BY => X"000000000001", -- Count by value
   DEVICE => "VIRTEX6",         -- Target Device: "VIRTEX5", "VIRTEX6", "SPARTAN6"
   WIDTH_DATA => 48)            -- Counter output bus width, 1-48
port map (
   Q => Q,                    -- Counter output, 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
//                     Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

COUNTER_LOAD_MACRO #(
   .COUNT_BY(48'h000000000001), // Count by value
   .DEVICE("VIRTEX6"), // 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
);
```

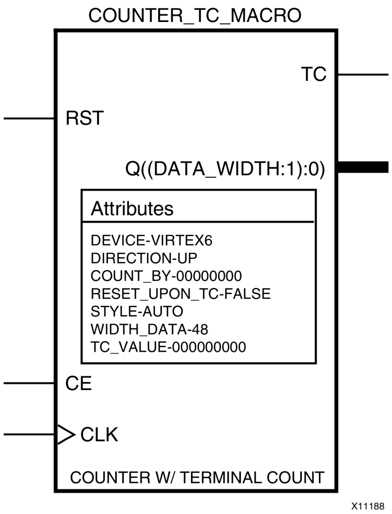
```
// End of COUNTER_LOAD_MACRO_inst instantiation
```

For More Information

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

COUNTER_TC_MACRO

Macro: Counter with Terminal Count



Introduction

The COUNTER_TC_MACRO simplifies the instantiation of the DSP48 block when used as a terminal count, up/down counter. It features parameterizable output width, terminal count values, count by and count direction in order to ease the integration of DSP48 block into HDL.

Port Description

Name	Direction	Width (Bits)	Function
Output Ports			
TC	Output	1	Terminal count goes high when TC_VALUE is reached
Q	Output	Variable, see WIDTH_DATA attribute.	Counter output
Input Ports			
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
RESET_UPON_TC	Boolean	True, False	False	Specifies whether to reset the counter upon reaching terminal count
DEVICE	String	"VIRTEX6", "SPARTAN6"	"VIRTEX6"	Target hardware architecture.
DIRECTION	String	"UP", "DOWN"	"UP"	Count up versus count down.
COUNT_BY	Hexa-decimal	Any 48 bit value	000000000001	Count by *n*; takes precedence over WIDTH_DATA
TC_VALUE	Hexa-decima	Any 48 bit value	All zeros	Terminal count value.
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_TC_MACRO: Counter with terminal count implemented in a DSP48E
--                   Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

COUNTER_TC_MACRO_inst : COUNTER_TC_MACRO
generic map (
   COUNT_BY => X"000000000001", -- Count by value
   DEVICE => "VIRTEX6",         -- Target Device: "VIRTEX5", "VIRTEX6"
   DIRECTION => "UP",            -- Counter direction "UP" or "DOWN"
   RESET_UPON_TC => "FALSE",      -- Reset counter upon terminal count, TRUE or FALSE
   TC_VALUE => X"000000000000", -- Terminal count value
   WIDTH_DATA => 48)            -- Counter output bus width, 1-48
port map (
   Q => Q,        -- Counter output, width determined by WIDTH_DATA generic
   TC => TC,      -- 1-bit terminal count output, high = terminal count is reached
   CLK => CLK,    -- 1-bit clock input
   CE => CE,      -- 1-bit clock enable input
   RST => RST       -- 1-bit active high synchronous reset
);
-- End of COUNTER_TC_MACRO_inst instantiation
```

Verilog Instantiation Template

```
// COUNTER_TC_MACRO: Counter with terminal count implemented in a DSP48E
//                   Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

COUNTER_TC_MACRO #(
   .COUNT_BY(48'h000000000001), // Count by value
   .DEVICE("VIRTEX6"),          // Target Device: "VIRTEX5", "VIRTEX6"
   .DIRECTION("UP"),            // Counter direction, "UP" or "DOWN"
   .RESET_UPON_TC("FALSE"), // Reset counter upon terminal count, "TRUE" or "FALSE"
   .TC_VALUE(48'h000000000000), // Terminal count value
   .WIDTH_DATA(48)              // Counter output bus width, 1-48
) COUNTER_TC_MACRO_inst (
   .Q(Q),     // Counter output bus, width determined by WIDTH_DATA parameter
   .TC(TC),   // 1-bit terminal count output, high = terminal count is reached
   .CLK(CLK), // 1-bit positive edge clock input
   .CE(CE),   // 1-bit active high clock enable input
   .RST(RST)  // 1-bit active high synchronous reset
);
```

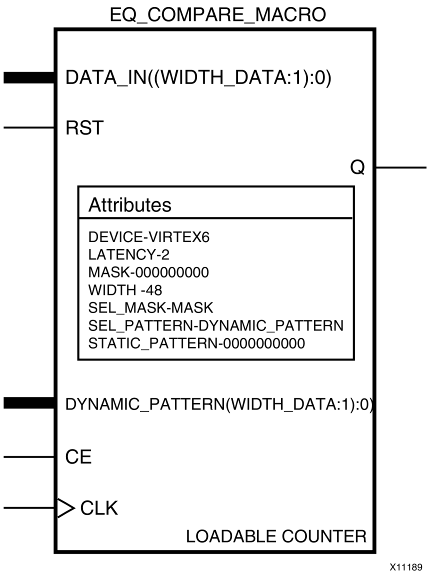
```
// End of COUNTER_TC_MACRO_inst instantiation
```

For More Information

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

EQ_COMPARE_MACRO

Macro: Equality Comparator



Introduction

The EQ_COMPARE_MACRO simplifies the instantiation of the DSP48 block when used as an equality comparator. It features parameterizable input and output widths, latencies, mask, and input sources that ease the integration of the DSP48 block into HDL.

Port Description

Name	Direction	Width	Function
Output Ports			
Q	Output	1	Active High pattern detection. Detects match of DATA_IN and the selected DYNAMIC_PATTERN gated by the MASK. Result arrives on the same cycle as P.
Input Ports			
DATA_IN	Input	Variable width, equals the value of the WIDTH attribute.	Input data to be compared
DYNAMIC_PATTERN	Input	Variable width, equals the value of the WIDTH attribute.	Dynamic data to be compared to DATA_IN
CLK	Input	1	Clock
CE	Inupt	1	Clock enable
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	"VIRTEX5", "VIRTEX6", "SPARTAN6"	"VIRTEX6"	Target hardware architecture.
SEL_PATTERN	Integer	1 to 24	24	Controls the width of PREADD1 and PREADD2 inputs.
MASK	Hexa-decimal	48 hex	all zeros	Mask to be used for pattern detector.
STATIC_PATTERN	Hexa-decimal	48 hex	all zeros	Pattern to be used for pattern detector.
SEL_MASK	String	"MASK", "DYNAMIC_ PATTERN"	"MASK"	Selects whether to use the static MASK or the C input for the mask of the pattern detector.
WIDTH	Integer	1 to 48	48	Width of DATA_IN and DYNAMIC_PATTERN
LATENCY	Integer	0, 1, 2, 3	2	Number of pipeline registers • 1: QREG == 1 • 2: AREG == BREG == CREG == QREG == 1 • 3: AREG == BREG == 2 and CREG == QREG == 1

VHDL Instantiation Template

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

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


-- EQ_COMPARE_MACRO: Equality Comparator implemented in a DSP48E
--                   Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

EQ_COMPARE_MACRO_inst : EQ_COMPARE_MACRO
generic map (
   DEVICE => "VIRTEX6",          -- Target Device: "VIRTEX5", "VIRTEX6"
   LATENCY => 2,                 -- Desired clock cycle latency, 0-2
   MASK => X"000000000000",      -- Select bits to be masked, must set
                                 -- SEL_MASK = "MASK"
   SEL_MASK => "MASK",           -- "MASK" = use MASK generic,
                                 -- "DYNAMIC_PATTERN = use DYNAMIC_PATTERN input bus
   SEL_PATTERN => "DYNAMIC_PATTERN", -- "DYNAMIC_PATTERN" = use DYNAMIC_PATTERN input bus
                                     -- "STATIC_PATTERN" = use STATIC_PATTERN generic
   STATIC_PATTERN => X"000000000000", -- Specify static pattern,
                                      -- must set SEL_PATTERN = "STATIC_PATTERN
   WIDTH => 48)             -- Comparator output bus width, 1-48
port map (
   Q => Q,        -- 1-bit output indicating a match
   CE => CE,      -- 1-bit active high input clock enable input
```

```
   CLK => CLK,    -- 1-bit positive edge clock input
   DATA_IN => DATA_IN, -- Input Data Bus, width determined by WIDTH generic
   DYNAMIC_PATTERN, => DYNAMIC_PATTERN, -- Input Dynamic Match/Mask Bus, width determined by WIDTH generic
   RST => RST       -- 1-bit input active high reset
);
-- End of EQ_COMPARE_MACRO_inst instantiation
```

Verilog Instantiation Template

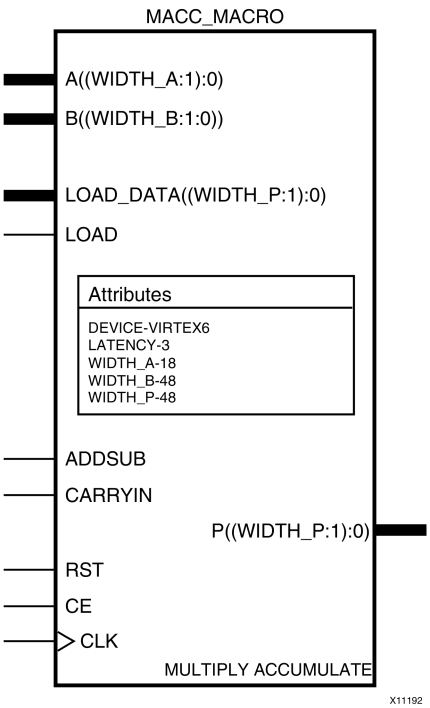
```
// EQ_COMPARE_MACRO: Equality Comparator implemented in a DSP48E
//                   Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

EQ_COMPARE_MACRO #(
   .DEVICE("VIRTEX6"),       // Target Device: "VIRTEX5", "VIRTEX6"
   .LATENCY(2),              // Desired clock cycle latency, 0-2
   .MASK(48'h000000000000),  // Select bits to be masked, must set SEL_MASK="MASK"
   .SEL_MASK("MASK"),        // "MASK" = use MASK parameter,
                             //   "DYNAMIC_PATTERN" = use DYNAMIC_PATTERN input bus
   .SEL_PATTERN("STATIC_PATTERN"), // "STATIC_PATTERN" = use STATIC_PATTERN parameter,
                                   //   "DYNAMIC_PATTERN = use DYNAMIC_PATTERN input bus
   .STATIC_PATTERN(48'h000000000000), // Specify static pattern, must set SEL_PATTERN = "STATIC_PATTERN"
   .WIDTH(48)                // Comparator output bus width, 1-48
) EQ_COMPARE_MACRO_inst (
   .Q(Q),     // 1-bit output indicating a match
   .CE(CE),   // 1-bit active high input clock enable
   .CLK(CLK), // 1-bit positive edge clock input
   .DATA_IN(DATA_IN), // Input Data Bus, width determined by WIDTH parameter
   .DYNAMIC_PATTERN(DYNAMIC_PATTERN), // Input Dynamic Match/Mask Bus, width determined by WIDTH parameter
   .RST(RST)  // 1-bit input active high reset
);

// End of EQ_COMPARE_MACRO_inst instantiation
```

For More Information

See the *Virtex-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 25	25	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
--             Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

MACC_MACRO_inst : MACC_MACRO
generic map (
   DEVICE => "VIRTEX6",  -- 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 output 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

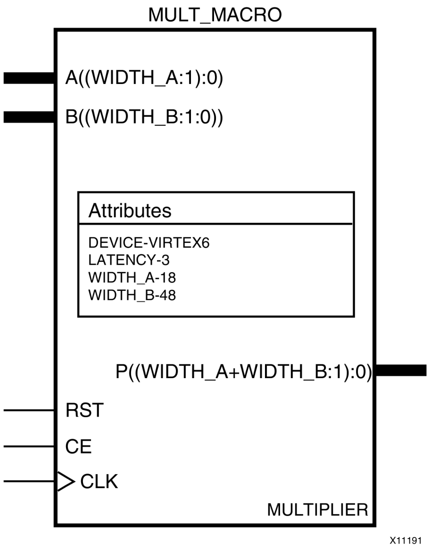
```
// MACC_MACRO: Multiply Accumulate Function implemented in a DSP48E
//             Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

MACC_MACRO #(
   .DEVICE("VIRTEX6"), // 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
) 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 *Virtex-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 25	25	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
--             Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

MULT_MACRO_inst : MULT_MACRO
generic map (
   DEVICE => "VIRTEX6",    -- 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 output 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
//             Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

MULT_MACRO #(
   .DEVICE("VIRTEX6"), // 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
) 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
```

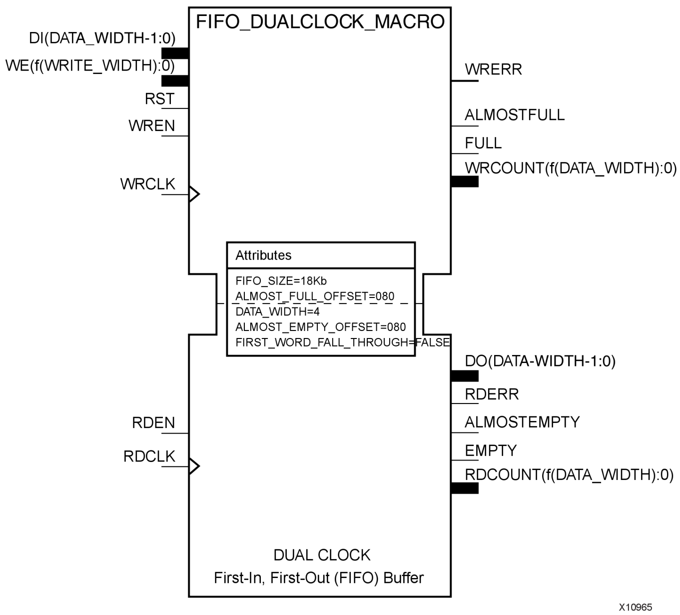
```
   .RST(RST)  // 1-bit input active high reset
);

// End of MULT_MACRO_inst instantiation
```

For More Information

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

FIFO_DUALCLOCK_MACRO

Macro: Dual Clock First-In, First-Out (FIFO) RAM Buffer



Introduction

FPGA devices contain several block RAM memories that can be configured as general-purpose 36kb or 18kb RAM/ROM memories. Dedicated logic in the block RAM enables you to easily implement FIFOs. The FIFO can be configured as an 18 kb or 36 kb memory. This unimacro configures the FIFO for using independent read and writes clocks. Data is read from the FIFO on the rising edge of read clock and written to the FIFO on the rising edge of write clock.

Depending on the offset between read and write clock edges, the Empty, Almost Empty, Full and Almost Full flags can deassert one cycle later. Due to the asynchronous nature of the clocks the simulation model only reflects the deassertion latency cycles listed in the User Guide.

Port Description

Name	Direction	Width	Function
Output Ports			
ALMOSTEMPTY	Output	1	Almost all valid entries in FIFO have been read.
ALMOSTFULL	Output	1	Almost all entries in FIFO memory have been filled.
DO	Output	See Configuration Table below.	Data output bus addressed by ADDR.
EMPTY	Output	1	FIFO is empty.
FULL	Output	1	All entries in FIFO memory are filled.
RDCOUNT	Output	See Configuration Table below.	FIFO data read pointer.
RDERR	Output	1	When the FIFO is empty, any additional read operation generates an error flag.

Name	Direction	Width	Function
WRCOUNT	Output	See Configuration Table below.	FIFO data write pointer.
WRERR	Output	1	When the FIFO is full, any additional write operation generates an error flag.
Input Ports			
DI	Input	See Configuration Table below.	Data input bus addressed by ADDR.
RDCLK	Input	1	Clock for Read domain operation.
RDEN	Input	1	Read Enable
RST	Input	1	Asynchronous reset.
WRCLK	Input	1	Clock for Write domain operation.
WREN	Input	1	Write Enable

Configuration Table

This unimacro can be instantiated only. The unimacro is a parameterizable version of the primitive. Please use the Configuration Table below to correctly configure the unimacro to meet design needs.

DATA_WIDTH	FIFO_SIZE	WRCOUNT	RDCOUNT
72 - 37	36kb	9	9
36 - 19	36kb	10	10
	18kb	9	9
18 - 10	36kb	11	11
	18kb	10	10
9-5	36kb	12	12
	18kb	11	11
1-4	36kb	13	13
	18kb	12	12

Design Entry Method

This unimacro can be instantiated only. It is a parameterizable version of the primitive. Consult the above Configuration Table to correctly configure 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
ALMOST_EMPTY_ OFFSET	Hexadecimal	13-Bit Value	All zeros	Setting determines the difference between EMPTY and ALMOSTEMPTY conditions. Must be set using hexadecimal notation.
ALMOST_FULL_ OFFSET	Hexadecimal	13-Bit Value	All zeros	Setting determines the difference between FULL and ALMOSTFULL conditions. Must be set using hexadecimal notation.
DATA_WIDTH	Integer	1 - 72	4	Width of DI/DO bus.
FIFO_SIZE	String	"18Kb", "36Kb"	"18Kb"	Configures FIFO as 18Kb or 36Kb memory.
FIRST_WORD_ FALL_THROUGH	Boolean	FALSE, TRUE	FALSE	If TRUE, the first word written into the empty FIFO appears at the FIFO output without RDEN asserted.
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.

VHDL Instantiation Template

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

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


-- FIFO_DUALCLOCK_MACRO: Dual-Clock First-In, First-Out (FIFO) RAM Buffer
--                       Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

-- 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 | FIFO_SIZE | FIFO Depth | RDCOUNT/WRCOUNT Width --
-- ===========|===========|============|=======================--
--   37-72    |  "36Kb"   |     512    |         9-bit         --
--   19-36    |  "36Kb"   |    1024    |        10-bit         --
--   19-36    |  "18Kb"   |     512    |         9-bit         --
--   10-18    |  "36Kb"   |    2048    |        11-bit         --
--   10-18    |  "18Kb"   |    1024    |        10-bit         --
--    5-9     |  "36Kb"   |    4096    |        12-bit         --
--    5-9     |  "18Kb"   |    2048    |        11-bit         --
--    1-4     |  "36Kb"   |    8192    |        13-bit         --
--    1-4     |  "18Kb"   |    4096    |        12-bit         --
-----------------------------------------------------------------


FIFO_DUALCLOCK_MACRO_inst : FIFO_DUALCLOCK_MACRO
generic map (
   DEVICE => "VIRTEX6",            -- Target Device: "VIRTEX5", "VIRTEX6"
   ALMOST_FULL_OFFSET => X"0080",  -- Sets almost full threshold
   ALMOST_EMPTY_OFFSET => X"0080", -- Sets the almost empty threshold
   DATA_WIDTH => 0,   -- Valid values are 1-72 (37-72 only valid when FIFO_SIZE="36Kb")
   FIFO_SIZE => "18Kb",            -- Target BRAM, "18Kb" or "36Kb"
   FIRST_WORD_FALL_THROUGH => FALSE) -- Sets the FIFO FWFT to TRUE or FALSE
port map (
   ALMOSTEMPTY => ALMOSTEMPTY,   -- 1-bit output almost empty
   ALMOSTFULL => ALMOSTFULL,     -- 1-bit output almost full
   DO => DO,                     -- Output data, width defined by DATA_WIDTH parameter
   EMPTY => EMPTY,               -- 1-bit output empty
   FULL => FULL,                 -- 1-bit output full
```

```
   RDCOUNT => RDCOUNT,           -- Output read count, width determined by FIFO depth
   RDERR => RDERR,               -- 1-bit output read error
   WRCOUNT => WRCOUNT,           -- Output write count, width determined by FIFO depth
   WRERR => WRERR,               -- 1-bit output write error
   DI => DI,                     -- Input data, width defined by DATA_WIDTH parameter
   RDCLK => RDCLK,               -- 1-bit input read clock
   RDEN => RDEN,                 -- 1-bit input read enable
   RST => RST,                   -- 1-bit input reset
   WRCLK => WRCLK,               -- 1-bit input write clock
   WREN => WREN                  -- 1-bit input write enable
);
-- End of FIFO_DUALCLOCK_MACRO_inst instantiation
```

Verilog Instantiation Template

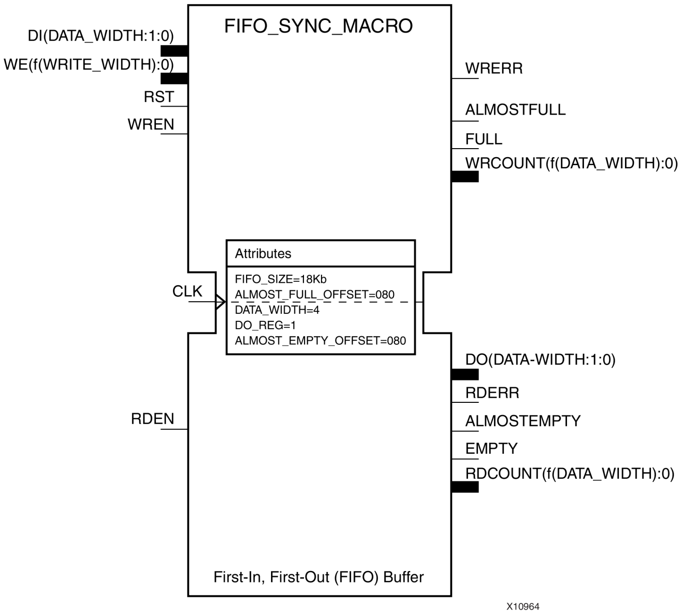
```
// FIFO_DUALCLOCK_MACRO: Dual Clock First-In, First-Out (FIfor) RAM Buffer
//                       Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

/////////////////////////////////////////////////////////////////
// DATA_WIDTH | FIFO_SIZE | FIfor Depth | RDCOUNT/WRCOUNT Width //
// ===========|===========|============|=======================//
//   37-72    |  "36Kb"   |     512    |         9-bit         //
//   19-36    |  "36Kb"   |    1024    |        10-bit         //
//   19-36    |  "18Kb"   |     512    |         9-bit         //
//   10-18    |  "36Kb"   |    2048    |        11-bit         //
//   10-18    |  "18Kb"   |    1024    |        10-bit         //
//    5-9     |  "36Kb"   |    4096    |        12-bit         //
//    5-9     |  "18Kb"   |    2048    |        11-bit         //
//    1-4     |  "36Kb"   |    8192    |        13-bit         //
//    1-4     |  "18Kb"   |    4096    |        12-bit         //
/////////////////////////////////////////////////////////////////

FIFO_DUALCLOCK_MACRO  #(
   .ALMOST_EMPTY_OFFSET(9'h080), // Sets the almost empty threshold
   .ALMOST_FULL_OFFSET(9'h080),  // Sets almost full threshold
   .DATA_WIDTH(0),   // Valid values are 1-72 (37-72 only valid when FIFO_SIZE="36Kb")
   .DEVICE("VIRTEX6"),  // Target device: "VIRTEX5", "VIRTEX6"
   .FIFO_SIZE ("18Kb"), // Target BRAM: "18Kb" or "36Kb"
   .FIRST_WORD_FALL_THROUGH ("FALSE") // Sets the FIfor FWFT to "TRUE" or "FALSE"
) FIFO_DUALCLOCK_MACRO_inst (
   .ALMOSTEMPTY(ALMOSTEMPTY), // 1-bit output almost empty
   .ALMOSTFULL(ALMOSTFULL),   // 1-bit output almost full
   .DO(DO),                   // Output data, width defined by DATA_WIDTH parameter
   .EMPTY(EMPTY),             // 1-bit output empty
   .FULL(FULL),               // 1-bit output full
   .RDCOUNT(RDCOUNT),         // Output read count, width determined by FIfor depth
   .RDERR(RDERR),             // 1-bit output read error
   .WRCOUNT(WRCOUNT),         // Output write count, width determined by FIfor depth
   .WRERR(WRERR),             // 1-bit output write error
   .DI(DI),                   // Input data, width defined by DATA_WIDTH parameter
   .RDCLK(RDCLK),             // 1-bit input read clock
   .RDEN(RDEN),               // 1-bit input read enable
   .RST(RST),                 // 1-bit input reset
   .WRCLK(WRCLK),             // 1-bit input write clock
   .WREN(WREN)                // 1-bit input write enable
);

// End of FIFO_DUALCLOCK_MACRO_inst instantiation
```

For More Information

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

FIFO_SYNC_MACRO

Macro: Synchronous First-In, First-Out (FIFO) RAM Buffer



Introduction

FPGA devices contain several block RAM memories that can be configured as general-purpose 36kb or 18kb RAM/ROM memories. Dedicated logic in the block RAM enables you to easily implement FIFOs. The FIFO can be configured as an 18 kb or 36 kb memory. This unimacro configures the FIFO such that it uses one clock for reading as well as writing.

Port Description

Name	Direction	Width	Function
Output Ports			
ALMOSTEMPTY	Output	1	Almost all valid entries in FIFO have been read.
ALMOSTFULL	Output	1	Almost all entries in FIFO memory have been filled.
DO	Output	See Configuration Table.	Data output bus addressed by ADDR.
EMPTY	Output	1	FIFO is empty.
FULL	Output	1	All entries in FIFO memory are filled.
RDCOUNT	Output	See Configuration Table below.	FIFO data read pointer.
RDERR	Output	1	When the FIFO is empty, any additional read operation generates an error flag.
WRCOUNT	Output	See Configuration Table.	FIFO data write pointer.
WRERR	Output	1	When the FIFO is full, any additional write operation generates an error flag.

Name	Direction	Width	Function
Input Ports			
CLK	Input	1	Clock for Read/Write domain operation.
DI	Input	See Configuration Table.	Data input bus addressed by ADDR.
RDEN	Input	1	Read Enable
RST	Input	1	Asynchronous reset.
WREN	Input	1	Write Enable

Configuration Table

This unimacro can be instantiated only. The unimacro is a parameterizable version of the primitive. Please use the configuration table below to correctly configure the unimacro to meet design needs.

DATA_WIDTH	FIFO_SIZE	WRCOUNT	RDCOUNT
72 - 37	36kb	9	9
36 - 19	36kb	10	10
	18kb	9	9
18 - 10	36kb	11	11
	18kb	10	10
9-5	36kb	12	12
	18kb	11	11
1-4	36kb	13	13
	18kb	12	12

Design Entry Method

This unimacro can be instantiated only. It is a parameterizable version of the primitive. Consult the above Configuration Table to correctly configure 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
ALMOST_EMPTY_ OFFSET	Hexadecimal	Any 13-bit Value	All zeros	Setting determines the difference between EMPTY and ALMOSTEMPTY conditions. Must be set using hexadecimal notation.
ALMOST_FULL_ OFFSET	Hexadecimal	Any 13-bit Value	All zeros	Setting determines the difference between FULL and ALMOSTFULL conditions. Must be set using hexadecimal notation.
DATA_WIDTH	Integer	1 - 72	4	Width of DI/DO bus.
DO_REG	Binary	0,1	1	DO_REG must be set to 0 for flags and data to follow a standard synchronous FIFO operation. When DO_REG is set to 1, effectively a pipeline register is added to the output of the synchronous FIFO. Data then has a one clock cycle latency. However, the clock-to-out timing is improved.
FIFO_SIZE	String	"18Kb", "36Kb"	"18Kb"	Configures FIFO as 18kb or 36kb memory.
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.

VHDL Instantiation Template

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

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


-- FIFO_SYNC_MACRO: Synchronous First-In, First-Out (FIFO) RAM Buffer
--                  Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

-- 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 | FIFO_SIZE | FIFO Depth | RDCOUNT/WRCOUNT Width --
-- ===========|===========|============|=======================--
--   37-72    |  "36Kb"   |     512    |         9-bit         --
--   19-36    |  "36Kb"   |    1024    |        10-bit         --
--   19-36    |  "18Kb"   |     512    |         9-bit         --
--   10-18    |  "36Kb"   |    2048    |        11-bit         --
--   10-18    |  "18Kb"   |    1024    |        10-bit         --
--    5-9     |  "36Kb"   |    4096    |        12-bit         --
--    5-9     |  "18Kb"   |    2048    |        11-bit         --
--    1-4     |  "36Kb"   |    8192    |        13-bit         --
--    1-4     |  "18Kb"   |    4096    |        12-bit         --
-----------------------------------------------------------------

FIFO_SYNC_MACRO_inst : FIFO_SYNC_MACRO
generic map (
   DEVICE => "VIRTEX6",            -- Target Device: "VIRTEX5, "VIRTEX6"
   ALMOST_FULL_OFFSET => X"0080",  -- Sets almost full threshold
   ALMOST_EMPTY_OFFSET => X"0080", -- Sets the almost empty threshold
   DATA_WIDTH => 0,   -- Valid values are 1-72 (37-72 only valid when FIFO_SIZE="36Kb")
   FIFO_SIZE => "18Kb)            -- Target BRAM, "18Kb" or "36Kb"
port map (
   ALMOSTEMPTY => ALMOSTEMPTY,   -- 1-bit output almost empty
   ALMOSTFULL => ALMOSTFULL,     -- 1-bit output almost full
```

```
   DO => DO,                     -- Output data, width defined by DATA_WIDTH parameter
   EMPTY => EMPTY,               -- 1-bit output empty
   FULL => FULL,                 -- 1-bit output full
   RDCOUNT => RDCOUNT,           -- Output read count, width determined by FIFO depth
   RDERR => RDERR,               -- 1-bit output read error
   WRCOUNT => WRCOUNT,           -- Output write count, width determined by FIFO depth
   WRERR => WRERR,               -- 1-bit output write error
   CLK => CLK,                   -- 1-bit input clock
   DI => DI,                     -- Input data, width defined by DATA_WIDTH parameter
   RDEN => RDEN,                 -- 1-bit input read enable
   RST => RST,                   -- 1-bit input reset
   WREN => WREN                  -- 1-bit input write enable
);
-- End of FIFO_SYNC_MACRO_inst instantiation
```

Verilog Instantiation Template

```
// FIFO_SYNC_MACRO: Synchronous First-In, First-Out (FIfor) RAM Buffer
//                  Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

/////////////////////////////////////////////////////////////////
// DATA_WIDTH | FIFO_SIZE | FIfor Depth | RDCOUNT/WRCOUNT Width //
// ===========|===========|============|=======================//
//   37-72    |  "36Kb"   |     512    |         9-bit         //
//   19-36    |  "36Kb"   |    1024    |        10-bit         //
//   19-36    |  "18Kb"   |     512    |         9-bit         //
//   10-18    |  "36Kb"   |    2048    |        11-bit         //
//   10-18    |  "18Kb"   |    1024    |        10-bit         //
//    5-9     |  "36Kb"   |    4096    |        12-bit         //
//    5-9     |  "18Kb"   |    2048    |        11-bit         //
//    1-4     |  "36Kb"   |    8192    |        13-bit         //
//    1-4     |  "18Kb"   |    4096    |        12-bit         //
/////////////////////////////////////////////////////////////////

FIFO_SYNC_MACRO  #(
   .DEVICE("VIRTEX6"), // Target Device: "VIRTEX5", "VIRTEX6"
   .ALMOST_EMPTY_OFFSET(9'h080), // Sets the almost empty threshold
   .ALMOST_FULL_OFFSET(9'h080),  // Sets almost full threshold
   .DATA_WIDTH(0), // Valid values are 1-72 (37-72 only valid when FIFO_SIZE="36Kb")
   .DO_REG(0),     // Optional output register (0 or 1)
   .FIFO_SIZE ("18Kb") // Target BRAM: "18Kb" or "36Kb"
) FIFO_SYNC_MACRO_inst (
   .ALMOSTEMPTY(ALMOSTEMPTY), // 1-bit output almost empty
   .ALMOSTFULL(ALMOSTFULL),   // 1-bit output almost full
   .DO(DO),                   // Output data, width defined by DATA_WIDTH parameter
   .EMPTY(EMPTY),             // 1-bit output empty
   .FULL(FULL),               // 1-bit output full
   .RDCOUNT(RDCOUNT),         // Output read count, width determined by FIfor depth
   .RDERR(RDERR),             // 1-bit output read error
   .WRCOUNT(WRCOUNT),         // Output write count, width determined by FIfor depth
   .WRERR(WRERR),             // 1-bit output write error
   .CLK(CLK),                 // 1-bit input clock
   .DI(DI),                   // Input data, width defined by DATA_WIDTH parameter
   .RDEN(RDEN),               // 1-bit input read enable
   .RST(RST),                 // 1-bit input reset
   .WREN(WREN)                // 1-bit input write enable
);

// End of FIFO_SYNC_MACRO_inst instantiation
```

For More Information

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

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
PCIE_2_0	Primitive: PCI Express version 2.0 compliant port
SYSMON	Primitive: System Monitor

Arithmetic Functions

Design Element	Description
DSP48E1	Primitive: 25x18 Two's Complement Multiplier with Integrated 48-Bit, 3-Input Adder/Subtracter/Accumulator or 2-Input Logic Unit

Clock Components

Design Element	Description
BUFG	Primitive: Global Clock Buffer
BUFGCE	Primitive: Global Clock Buffer with Clock Enable
BUFGCE_1	Primitive: Global Clock Buffer with Clock Enable and Output State 1
BUFGCTRL	Primitive: Global Clock MUX Buffer
BUFGMUX	Primitive: Global Clock MUX Buffer
BUFGMUX_1	Primitive: Global Clock MUX Buffer with Output State 1
BUFGMUX_CTRL	Primitive: 2-to-1 Global Clock MUX Buffer
BUFH	Primitive: Clock buffer for a single clocking region

Design Element	Description
BUFHCE	Primitive: Clock buffer for a single clocking region with clock enable
BUFIO	Primitive: Local Clock Buffer for I/O
BUFIODQS	Primitive: Differential Clock Input for Transceiver Reference Clocks
BUFR	Primitive: Regional Clock Buffer for I/O and Logic Resources
IBUFDS_GTXE1	Primitive: Differential Clock Input for the Transceiver Reference Clocks
MMCM_ADV	Primitive: MMCM is a mixed signal block designed to support clock network deskew, frequency synthesis, and jitter reduction.
MMCM_BASE	Primitive: Mixed signal block designed to support clock network deskew, frequency synthesis, and jitter reduction.

Config/BSCAN Components

Design Element	Description
BSCAN_VIRTEX6	Primitive: Virtex®-6 JTAG Boundary-Scan Logic Access Circuit
CAPTURE_VIRTEX6	Primitive: Virtex®-6 Readback Register Capture Control
DNA_PORT	Primitive: Device DNA Data Access Port
EFUSE_USR	Primitive: 32-bit non-volatile design ID
FRAME_ECC_VIRTEX6	Primitive: Virtex®-6 Configuration Frame Error Detection and Correction Circuitry
ICAP_VIRTEX6	Primitive: Internal Configuration Access Port
JTAG_SIM_VIRTEX6	Simulation: JTAG TAP Controller Simulation Model
SIM_CONFIG_V6	Simulation: Configuration Simulation Model
SIM_CONFIG_V6_SERIAL	Simulation: Serial Configuration Simulation Model
STARTUP_VIRTEX6	Primitive: Virtex®-6 Configuration Start-Up Sequence Interface
USR_ACCESS_VIRTEX6	Primitive: Virtex-6 User Access Register

I/O Components

Design Element	Description
DCIRESET	Primitive: DCI State Machine Reset (After Configuration Has Been Completed)
GTHE1_QUAD	Primitive: Gigabit Transceiver
GTXE1	Primitive: Gigabit Transceiver
IBUF	Primitive: Input Buffer
IBUFDS	Primitive: Differential Signaling Input Buffer
IBUFDS_DIFF_OUT	Primitive: Signaling Input Buffer with Differential Output
IBUFDS_GTHE1	Primitive: Differential Clock Input for the GTH Transceiver Reference Clocks

Design Element	Description
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
IDELAYCTRL	Primitive: IDELAY Tap Delay Value Control
IOBUF	Primitive: Bi-Directional Buffer
IOBUFDS	Primitive: 3-State Differential Signaling I/O Buffer with Active Low Output Enable
IODELAYE1	Primitive: Input and Output Fixed or Variable Delay Element
ISERDESE1	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
OSERDESE1	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
TEMAC_SINGLE	Primitive: Tri-mode Ethernet Media Access Controller (MAC)

RAM/ROM

Design Element	Description
FIFO18E1	Primitive: 18 k-bit FIFO (First In, First Out) Block RAM Memory
FIFO36E1	Primitive: 36 kb FIFO (First In, First Out) Block RAM Memory
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

Design Element	Description
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
RAMB18E1	Primitive: 18K-bit Configurable Synchronous Block RAM
RAMB36E1	Primitive: 36K-bit Configurable Synchronous Block RAM
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
IDDR	Primitive: Input Dual Data-Rate Register
IDDR_2CLK	Primitive: Input Dual Data-Rate Register with Dual Clock Inputs
LDCE	Primitive: Transparent Data Latch with Asynchronous Clear and Gate Enable
LDPE	Primitive: Transparent Data Latch with Asynchronous Preset and Gate Enable
ODDR	Primitive: Dedicated Dual Data Rate (DDR) Output Register

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
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

Chapter 4

About Design Elements

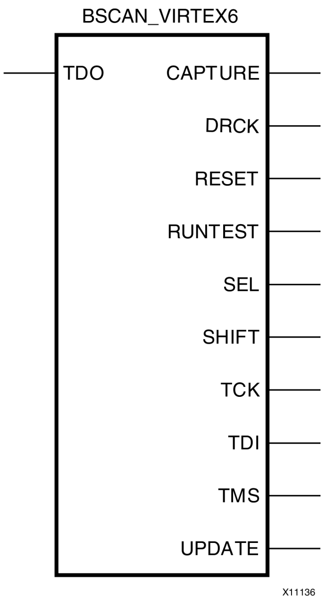
This section describes the design elements that can be used with Virtex®-6 devices. The design elements are organized alphabetically.

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_VIRTEX6

Primitive: Virtex®-6 JTAG Boundary-Scan Logic Access Circuit



Introduction

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

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 Programmable Logic Data Sheet for this element.

Port Descriptions

Port	Direction	Width	Function
CAPTURE	Output	1	Scan Data Register Capture instruction.
DRCK	Output	1	Scan Clock instruction. DRCK is a gated version of TCTCK, it toggles during the CAPTUREDR and SHIFTDR states.
RESET	Output	1	Scan register reset instruction.
RUNTEST	Output	1	Asserted when TAP controller is in Run Test Idle state.
SEL	Output	1	Scan mode Select instruction.
SHIFT	Output	1	Scan Chain Shift instruction.
TCK	Output	1	Scan Clock. Fabric connection to TAP Clock pin.
TDI	Output	1	Scan Chain Output. Mirror of TDI input pin to FPGA
TDO	Input	1	Scan Chain Input.
TMS	Output	1	Test Mode Select. Fabric connection to TAP.
UPDATE	Output	1	Scan Register Update instruction.

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_VIRTEX6: Boundary Scan
--                Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

BSCAN_VIRTEX6_inst : BSCAN_VIRTEX6
generic map (
   DISABLE_JTAG => FALSE, -- This attribute is unsupported. Please leave it at default.
   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: State output 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 input. 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_VIRTEX6_inst instantiation
```

Verilog Instantiation Template

```
// BSCAN_VIRTEX6: Boundary Scan
//                Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

BSCAN_VIRTEX6 #(
   .DISABLE_JTAG("FALSE"), // This attribute is unsupported. Please leave it at default.
   .JTAG_CHAIN(1)          // Value for USER command. Possible values: (1,2,3 or 4).
)
BSCAN_VIRTEX6_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: State output asserted when TAP controller is in Run Test Idle state.
   .SEL(SEL),         // 1-bit output: USER active output
```

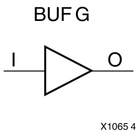
```
   .SHIFT(SHIFT),      // 1-bit output: SHIFT output from TAP controller
   .TCK(TCK),          // 1-bit output: Scan Clock output. Fabric connection to TAP Clock pin.
   .TDI(TDI),          // 1-bit output: TDI output from TAP controller
   .TMS(TMS),          // 1-bit output: Test Mode Select input. 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_VIRTEX6_inst instantiation
```

For More Information

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

BUFG

Primitive: Global Clock Buffer



Introduction

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

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
--       Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

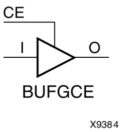
```
// BUFG: Global Clock Buffer
//       Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

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

BUFGCE

Primitive: Global Clock Buffer with Clock Enable



Introduction

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

Logic Table

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

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
--         Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

BUFGCE_inst : BUFGCE
port map (
   O => O,   -- 1-bit output: Clock buffer output
   CE => CE, -- 1-bit input: Clock enable input for I0 input
   I => I    -- 1-bit input: Primary clock input
);

-- End of BUFGCE_inst instantiation
```

Verilog Instantiation Template

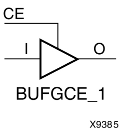
```
// BUFGCE: Global Clock Buffer with Clock Enable
//         Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

BUFGCE BUFGCE_inst (
   .O(O),   // 1-bit output: Clock buffer output
   .CE(CE), // 1-bit input: Clock enable input for I0 input
   .I(I)    // 1-bit input: Primary clock input
);

// End of BUFGCE_inst instantiation
```

For More Information

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

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
--             Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

BUFGCE_1_inst : BUFGCE_1
port map (
   O => O,    -- 1-bit output: Clock buffer output
   CE => CE,  -- 1-bit input: Clock enable input for I0 input
   I => I     -- 1-bit input: Primary clock input
);
```

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

Verilog Instantiation Template

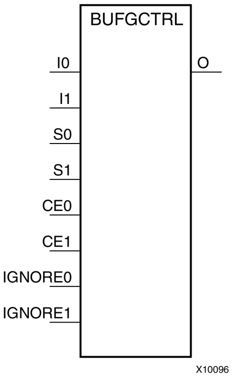
```
// BUFGCE_1: Global Clock Buffer with Clock Enable and Output State 1
//           Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

BUFGCE_1 BUFGCE_1_inst (
   .O(O),   // 1-bit output: Clock buffer output
   .CE(CE), // 1-bit input: Clock enable input for I0 input
   .I(I)    // 1-bit input: Primary clock input
);

// End of BUFGCE_1_inst instantiation
```

For More Information

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

BUFGCTRL

Primitive: Global Clock MUX Buffer



Introduction

BUFGCTRL primitive is global clock buffer that is designed as a synchronous/asynchronous "glitch free" 2:1 multiplexer with two clock inputs. Unlike global clock buffers that are found in previous generation of FPGAs, these clock buffers are designed with more control pins to provide a wider range of functionality and more robust input switching. BUFGCTRL is not limited to clocking applications.

Port Descriptions

Port	Direction	Width	Function
O	Output	1	Clock Output pin
I0, I1	Input	1 (each)	Clock Input: • I0 - Clock Input Pin • I1 - Clock Input Pin
CE0, CE1	Input	1 (each)	Clock Enable Input. The CE pins represent the clock enable pin for each clock inputs and are used to select the clock inputs. A setup/hold time must be specified when you are using the CE pin to select inputs. Failure to meet this requirement could result in a clock glitch.
S0, S1	Input	1 (each)	Clock Select Input. The S pins represent the clock select pin for each clock inputs. When using the S pin as input select, there is a setup/hold time requirement. Unlike CE pins, failure to meet this requirement will not result in a clock glitch. However, it can cause the output clock to appear one clock cycle later.
IGNORE0, IGNORE1	Input	1 (each)	Clock Ignore Input. IGNORE pins are used whenever a designer wants to bypass the switching algorithm executed by the BUFGCTRL.

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_OUT	Integer	0, 1	0	Initializes the BUFGCTRL output to the specified value after configuration.
PRESELECT_I0	Boolean	FALSE, TRUE	FALSE	If TRUE, BUFGCTRL output uses I0 input after configuration.
PRESELECT_I1	Boolean	FALSE, TRUE	FALSE	If TRUE, BUFGCTRL output uses I1 input after configuration.

Note Both PRESELECT attributes might not be TRUE at the same time.

VHDL Instantiation Template

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

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


-- BUFGCTRL: Global Clock MUX Buffer
--           Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

BUFGCTRL_inst : BUFGCTRL
generic map (
   INIT_OUT => 0,         -- Initial value of BUFGCTRL output (0/1)
   PRESELECT_I0 => FALSE, -- BUFGCTRL output uses I0 input (TRUE/FALSE)
   PRESELECT_I1 => FALSE  -- BUFGCTRL output uses I1 input (TRUE/FALSE)
)
port map (
   O => O,               -- 1-bit output: Clock Output pin
   CE0 => CE0,           -- 1-bit input: Clock enable input for I0 input
   CE1 => CE1,           -- 1-bit input: Clock enable input for I1 input
   I0 => I0,             -- 1-bit input: Primary clock input
   I1 => I1,             -- 1-bit input: Secondary clock input
   IGNORE0 => IGNORE0,   -- 1-bit input: Clock ignore input for I0
   IGNORE1 => IGNORE1,   -- 1-bit input: Clock ignore input for I1
   S0 => S0,             -- 1-bit input: Clock select input for I0
   S1 => S1              -- 1-bit input: Clock select input for I1
);

-- End of BUFGCTRL_inst instantiation
```

Verilog Instantiation Template

```
// BUFGCTRL: Global Clock MUX Buffer
//           Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

BUFGCTRL #(
   .INIT_OUT(0),           // Initial value of BUFGCTRL output (0/1)
   .PRESELECT_I0("FALSE"), // BUFGCTRL output uses I0 input (TRUE/FALSE)
   .PRESELECT_I1("FALSE")  // BUFGCTRL output uses I1 input (TRUE/FALSE)
)
BUFGCTRL_inst (
   .O(O),             // 1-bit output: Clock Output pin
   .CE0(CE0),         // 1-bit input: Clock enable input for I0 input
   .CE1(CE1),         // 1-bit input: Clock enable input for I1 input
   .I0(I0),           // 1-bit input: Primary clock input
   .I1(I1),           // 1-bit input: Secondary clock input
   .IGNORE0(IGNORE0), // 1-bit input: Clock ignore input for I0
   .IGNORE1(IGNORE1), // 1-bit input: Clock ignore input for I1
   .S0(S0),           // 1-bit input: Clock select input for I0
   .S1(S1)            // 1-bit input: Clock select input for I1
```

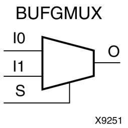
```
);
// End of BUFGCTRL_inst instantiation
```

For More Information

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

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

VHDL 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
--          Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

BUFGMUX_inst : BUFGMUX
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

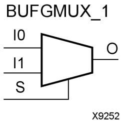
```
// BUFGMUX: Global Clock Mux Buffer
//          Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

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

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
--            Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

BUFGMUX_1_inst : BUFGMUX_1
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_1_inst instantiation
```

Verilog Instantiation Template

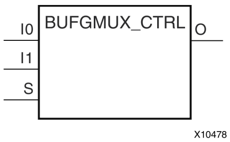
```
// BUFGMUX_1: Global Clock Mux Buffer with Output State 1
//            Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

BUFGMUX_1 BUFGMUX_1_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_1_inst instantiation
```

For More Information

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

BUFGMUX_CTRL

Primitive: 2-to-1 Global Clock MUX Buffer



Introduction

This design element is a global clock buffer with two clock inputs, one clock output, and a select line used to cleanly select between one of two clocks driving the global clocking resource. This component is based on BUFGCTRL, with some pins connected to logic High or Low. This element uses the S pin as the select pin for the 2-to-1 MUX. S can switch anytime without causing a glitch on the output clock of the buffer.

Port Descriptions

Port	Direction	Width	Function
O	Output	1	Clock Output
I0	Input	1	One of two Clock Inputs
I1	Input	1	One of two Clock Inputs
S0:S1	Input	1 each	Clock Select Input. The S pins represent the clock select pin for each clock input. When using the S pins as input select, there is a setup/hold time requirement. Unlike CE pins, failure to meet this requirement does not result in a clock glitch. However, it can cause the output clock to appear one clock cycle later.

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_CTRL: 2-to-1 Global Clock MUX Buffer
--                Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

BUFGMUX_CTRL_inst : BUFGMUX_CTRL
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_CTRL_inst instantiation
```

Verilog Instantiation Template

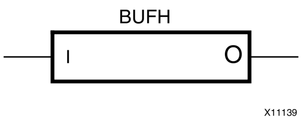
```
// BUFGMUX_CTRL: 2-to-1 Global Clock MUX Buffer
//                Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

BUFGMUX_CTRL BUFGMUX_CTRL_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_CTRL_inst instantiation
```

For More Information

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

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 *Virtex-6 FPGA Clocking Resources User Guide (UG362)* 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
--       Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

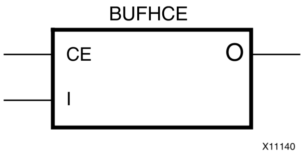
```
// BUFH: HROW Clock Buffer for a Single Clocking Region
//       Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

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

BUFHCE

Primitive: Clock buffer for a single clocking region with clock enable



Introduction

This element is provided to allow instantiation access to HCLK clock buffer resources. In addition, it allows for power reduction capabilities through disabling of the clock via clock enable (CE).

Port Descriptions

Port	Direction	Width	Function
CE	Input	1	Enables propagation of signal from I to O. When low, sets output to 0.
I	Input	1	The input to the BUFH
O	Output	1	The output of the BUFH

Design Entry Method

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

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
INIT_OUT	DECIMAL	0, 1	0	Initial output value. Also indicates Stop Low vs Stop High 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;


-- BUFHCE: HROW Clock Buffer for a Single Clocking Region with Clock Enable
--         Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

BUFHCE_inst : BUFHCE
generic map (
   INIT_OUT => 0  -- Initial output value
)
port map (
   O => O,   -- 1-bit output: Clock output
   CE => CE, -- 1-bit input: Active high enable input
   I => I    -- 1-bit input: Clock input
```

```
);

-- End of BUFHCE_inst instantiation
```

Verilog Instantiation Template

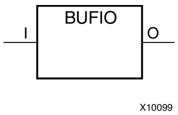
```
// BUFHCE: HROW Clock Buffer for a Single Clocking Region with Clock Enable
//         Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

BUFHCE #(
   .INIT_OUT(0)  // Initial output value
)
BUFHCE_inst (
   .O(O),    // 1-bit output: Clock output
   .CE(CE), // 1-bit input: Active high enable input
   .I(I)    // 1-bit input: Clock input
);

// End of BUFHCE_inst instantiation
```

For More Information

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

BUFIO

Primitive: Local Clock Buffer for I/O



Introduction

This design element is a clock buffer. It is simply a clock-in, clock-out buffer. It drives a dedicated clock net within the I/O column, independent of the global clock resources. Thus, these elements are ideally suited for source-synchronous data capture (forwarded/receiver clock distribution). They can only be driven by clock capable I/Os located in the same clock region. They drive the two adjacent I/O clock nets (for a total of up to three clock regions), as well as the regional clock buffers (BUFR). These elements cannot drive logic resources (CLB, block RAM, etc.) because the I/O clock network only reaches the I/O column.

Port Descriptions

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

Design Entry Method

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

VHDL Instantiation Template

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

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


-- BUFIO: Local Clock Buffer for I/O
--        Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

BUFIO_inst : BUFIO
port map (
   O => O, -- 1-bit output: Clock output port (connect to I/O clock loads)
   I => I  -- 1-bit input: Clock input port (connect to IBUFG)
);

-- End of BUFIO_inst instantiation
```

Verilog Instantiation Template

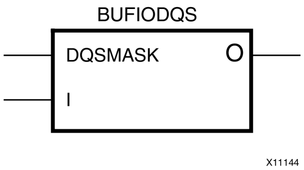
```
// BUFIO: Local Clock Buffer
//        Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

BUFIO BUFIO_inst (
   .O(O),     // Clock buffer output
   .I(I)      // Clock buffer input
);

// End of BUFIO_inst instantiation
```

For More Information

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

BUFIODQS

Primitive: Differential Clock Input for Transceiver Reference Clocks



Introduction

This element is the same clock buffer as BUFIO with added dedicated circuitry (ideally used for memory applications) to optionally remove the extra BUFIO delay and also squelch the I/O Clock after a given burst length from the strobe. In general, this component should only be used with the Xilinx® Memory Interface Generator (MIG) product.

Port Descriptions

Port	Direction	Width	Function
DQSMASK	Input	1	"Squelch" the I/O clock after a given burst length from strobe.
I	Input	1	Clock input port.
O	Output	1	Clock output port.

Design Entry Method

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

Available Attributes

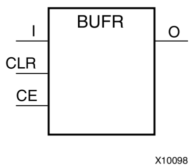
Attribute	Data Type	Allowed Values	Default	Description
DQSMASK_ENABLE	Boolean	FALSE, TRUE	FALSE	Enables the squelch circuitry

For More Information

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

BUFR

Primitive: Regional Clock Buffer for I/O and Logic Resources



Introduction

The regional clock buffer (BUFR) is another available clock buffer. BUFRs drive clock signals to a dedicated clock net within a clock region, independent from the global clock tree. Each BUFR can drive the six regional clock nets in the region where it is located, and the six clock nets in the adjacent clock regions (up to three clock regions). Unlike BUFIOs, BUFRs can drive the I/O logic and logic resources (CLB, block RAM, etc.) in the existing and adjacent clock regions. BUFRs can be driven by clock-capable pins, local interconnect, GTs, and the MMCMs high-performance clocks. In addition, BUFR is capable of generating divided clock outputs with respect to the clock input. The divide values are an integer between one and eight. BUFRs are ideal for source-synchronous applications requiring clock domain crossing or serial-to-parallel conversion. Each I/O column supports regional clock buffers. There are up to four I/O columns in a device with two inner columns (center left and right) and up to two outer left and right columns. The availability of the outer columns are device dependant while the inner columns are always present. The Virtex®-6 architecture therefore can have up to four BUFRs per region with two driving from the inner columns out (always present), and two BUFRs per region driving from the outer I/O columns in (when present). In Virtex-6 devices, BUFRs can also directly drive MMCM clock inputs and BUFGs.

Port Descriptions

Port	Direction	Width	Function
CE	Input	1	Clock enable port. When asserted low, this port disables the output clock. When asserted high, the clock is propagated out the O output port. Cannot be used in "BYPASS" mode. Connect to vcc when BUFR_DIVIDE is set to "BYPASS" or if not used.
CLR	Input	1	Counter asynchronous clear for divided clock output. When asserted high, this port resets the counter used to produce the divided clock output and the output is asserted low. Cannot be used in "BYPASS" mode. Connect to gnd when BUFR_DIVIDE is set to "BYPASS" or if not used.
I	Input	1	Clock input port. This port is the clock source port for BUFR. It can be driven by BUFIO output or local interconnect.
O	Output	1	Clock output port. This port drives the clock tracks in the clock region of the BUFR and the two adjacent clock regions. This port drives FPGA fabric, and IOBs.

Design Entry Method

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

Available Attributes

Attribute	Data Type	Allowed_Values	Default	Description
BUFR_DIVIDE	String	"BYPASS", "1", "2", "3", "4", "5", "6", "7", "8"	"BYPASS"	Defines whether the output clock is a divided version of input clock.
SIM_DEVICE	String	"VIRTEX4", VIRTEX5", "VIRTEX6"	"VIRTEX4"	Determine the CE latency for BUFR.

VHDL Instantiation Template

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

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


-- BUFR: Regional Clock Buffer for I/O and Logic Resources within a Clock Region
--       Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

BUFR_inst : BUFR
generic map (
   BUFR_DIVIDE => "BYPASS",   -- "BYPASS", "1", "2", "3", "4", "5", "6", "7", "8"
   SIM_DEVICE => "VIRTEX6")   -- Must be set to "VIRTEX6"
port map (
   O => O,     -- 1-bit output: Clock output port
   CE => CE,   -- 1-bit input: Active high, clock enable (Divided modes only)
   CLR => CLR, -- 1-bit input: Active high, asynchronous clear (Divided mode only)
   I => I     -- 1-bit input: Clock buffer input driven by an IBUFG, MMCM or local interconnect
);

-- End of BUFR_inst instantiation
```

Verilog Instantiation Template

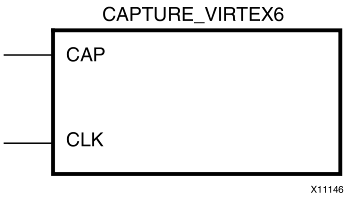
```
// BUFR: Regional Clock Buffer for I/O and Logic Resources within a Clock Region
//       Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

BUFR #(
   .BUFR_DIVIDE("BYPASS"), // Values: "BYPASS, 1, 2, 3, 4, 5, 6, 7, 8"
   .SIM_DEVICE("VIRTEX6")  // Must be set to "VIRTEX6"
)
BUFR_inst (
   .O(O),     // 1-bit output: Clock output port
   .CE(CE),   // 1-bit input: Active high, clock enable (Divided modes only)
   .CLR(CLR), // 1-bit input: Active high, asynchronous clear (Divided modes only)
   .I(I)      // 1-bit input: Clock buffer input driven by an IBUFG, MMCM or local interconnect
);

// End of BUFR_inst instantiation
```

For More Information

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

CAPTURE_VIRTEX6

Primitive: Virtex®-6 Readback Register Capture Control



Introduction

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

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

Port Descriptions

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

Design Entry Method

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

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

Available Attributes

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

VHDL Instantiation Template

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

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


-- CAPTURE_VIRTEX6: Register Capture
--                  Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

CAPTURE_VIRTEX6_inst : CAPTURE_VIRTEX6
generic map (
   ONESHOT => TRUE  -- Specifies the procedure for performing single readback per CAP trigger.
)
port map (
   CAP => CAP, -- 1-bit input: Capture Input
   CLK => CLK  -- 1-bit input: Clock Input
);

-- End of CAPTURE_VIRTEX6_inst instantiation
```

Verilog Instantiation Template

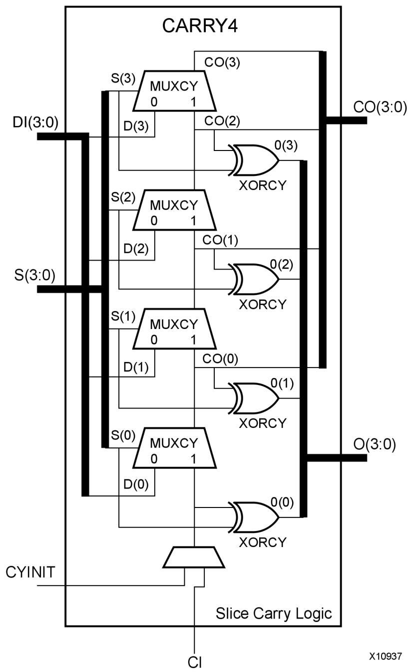
```
// CAPTURE_VIRTEX6: Register Capture
//                  Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

CAPTURE_VIRTEX6 #(
   .ONESHOT("TRUE")  // Specifies the procedure for performing single readback per CAP trigger.
)
CAPTURE_VIRTEX6_inst (
   .CAP(CAP), // 1-bit input: Capture Input
   .CLK(CLK)  // 1-bit input: Clock Input
);

// End of CAPTURE_VIRTEX6_inst instantiation
```

For More Information

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

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
--         Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

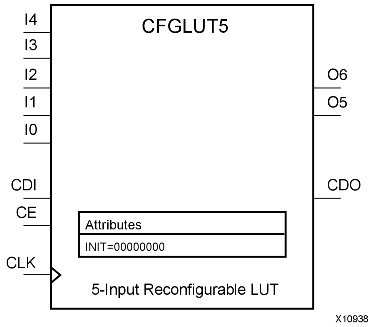
```
// CARRY4: Fast Carry Logic Component
//         Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

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

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
--          Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

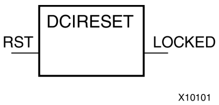
```
// CFGLUT5: Reconfigurable 5-input LUT
//          Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

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

DCIRESET

Primitive: DCI State Machine Reset (After Configuration Has Been Completed)



Introduction

This design element is used to reset the DCI state machine after configuration has been completed.

Port Descriptions

Port	Direction	Width	Function
LOCKED	Output	1	DCIRESET LOCK status output.
RST	Input	1	DCIRESET asynchronous reset input.

Design Entry Method

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

VHDL Instantiation Template

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

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


-- DCIRESET: Digitally Controlled Impedance Reset Component
--           Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

DCIRESET_inst : DCIRESET
port map (
   LOCKED => LOCKED, -- 1-bit output: LOCK status output
   RST => RST        -- 1-bit input: Active-high asynchronous reset input
);

-- End of DCIRESET_inst instantiation
```

Verilog Instantiation Template

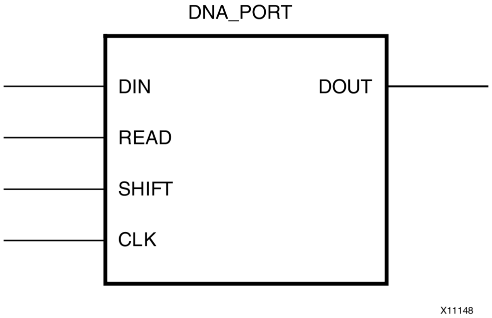
```
// DCIRESET: Digitally Controlled Impedance Reset Component
//           Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

DCIRESET DCIRESET_inst (
   .LOCKED(LOCKED), // 1-bit output: LOCK status output
   .RST(RST)        // 1-bit input: Active-high asynchronous reset input
);

// End of DCIRESET_inst instantiation
```

For More Information

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

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'h000000000000000 to 57'h1ffffffffffffff	57'h000000000000000	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 Access Port
--           Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

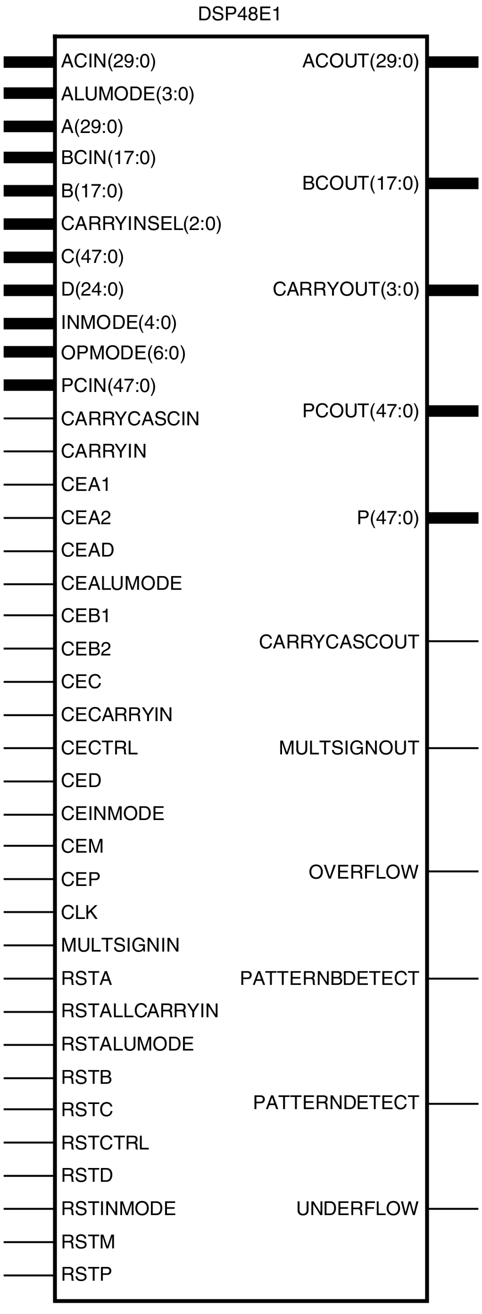
```
// DNA_PORT: Device DNA Access Port
//           Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

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

DSP48E1

Primitive: 25x18 Two's Complement Multiplier with Integrated 48-Bit, 3-Input Adder/Subtracter/Accumulator or 2-Input Logic Unit



Introduction

This design element is a versatile, scalable, hard IP block within Virtex®-6 that allows for the creation of compact, high-speed, arithmetic-intensive operations, such as those seen for many DSP algorithms. Some of the functions capable within the block include multiplication, addition (including pre-adder), subtraction, accumulation, shifting, logical operations, and pattern detection.

Port Descriptions

Port	Direction	Width	Function
A[29:0]	Input	30	25-bit data input to Multiplier, Pre-adder, or 30-bit MSB Data Input to Adder/Logic Unit. Tie port to all ones if not used.
ACIN[29:0]	Input	30	Cascade input for Port A. If used, connect to ACOUT of upstream cascaded DSP slice. Tie port to all zeros if not used.
ACOUT[29:0]	Output	30	Cascade output for Port A. If used, connect to ACIN of downstream cascaded DSP slice. Leave unconnected if not used.
ALUMODE[3:0]	Input	4	Control input to select Logic Unit functions including addition and subtraction.
B[17:0]	Input	18	18-bit data input to Multiplier, or 18-bit LSB Data Input to Adder/Logic Unit. Tie port to all ones if not used.
BCIN[17:0]	Input	18	Cascade input for Port B. If used, connect to BCOUT of upstream cascaded DSP slice. Tie port to all zeros if not used.
BCOUT[17:0]	Output	18	Cascade output for Port B. If used, connect to BCIN of downstream cascaded DSP slice. Leave unconnected if not used.
C[47:0]	Input	48	48-bit data input to Adder/Logic Unit and Pattern Detector. Tie port to all ones if not used.
CARRYCASCIN	Input	1	Cascaded CARRYIN from upstream DSP slice.
CARRYCASCOUT	Output	1	Cascaded CARRYOUT to downstream DSP slice.
CARRYIN	Input	1	External carry input to the Adder/Logic Unit.
CARRYINSEL[2:0]	Input	3	Selects carry-in source to the DSP slice.
CARRYOUT[3:0]	Output	4	Carry out signal for arithmetic operations (addition, subtraction, etc.). • If USE_SIMD="FOUR12", CARRYOUT represents the carry-out of each 12 bit field of the Accumulate/Adder/Logic Unit. • If USE_SIMD="TWO24" CARRYOUT and CARRYOUT represent the carry-out of each 24-bit field of the Accumulator/Adder. • If USE_SIMD="ONE48", CARRYOUT is the only valid carry out from the Accumulate/Adder/Logic Unit.
CEAD	Input	1	Active High, clock enable for pre-adder output AD pipeline register. Tie to logic one if not used and ADREG=1. Tie to logic zero if ADREG=0.
CEALUMODE	Input	1	Active High, clock enable for the ALUMODE input registers (ALUMODEREG=1). If ALUMODEREG=0, CEALUMODE should be tied to logic 0.
CEA1	Input	1	Active high, clock enable for the first A (input) register. This port is only used if AREG = 2 or INMODE0 = 1. Tie to logic one if not used and AREG=2. When two registers are used, this is the first sequentially. When Dynamic AB Access is used, this clock enable is applied for INMODE[0]=1. If A/ACIN port is not used, AREG should be set to 1 and CEA1 tied to 0.
CEA2	Input	1	Active High, clock enable for the A port registers. Tie to logic one if not used and AREG=1 or 2. Tie to logic zero if AREG=0. When two registers are used, this is the second sequentially.

Port	Direction	Width	Function
CEB1	Input	1	Active high, clock enable for the first B (input) register. This port is only used if BREG = 2 or INMODE0 = 1. Tie to logic one if not used and BREG=2. When two registers are used, this is the first sequentially. When Dynamic AB Access is used, this clock enable is applied for INMODE[0]=1. If B/BCIN port is not used, BREG should be set to 1 and CEB1 tied to 0.
CEB2	Input	1	Active High, clock enable for the B port registers. Tie to logic one if not used and BREG=1 or 2. Tie to logic zero if BREG=0. When two registers are used, this is the second sequentially.
CEC	Input	1	Active High, clock enable for the C port registers (CREG=1). If C port is not used, CREG should be set to 1 and CEC tied to logic 0.
CECARRYIN	Input	1	Active High, clock enable for the carry-in registers (CARRYINREG=1). If CARRYIN=0, CARRYINREG should be tied to logic 0.
CECTRL	Input	1	Active High, clock enable for the OPMODE and CARRYINSEL registers. If OPMODEREG=0, CARRYINSELREG should be tied to logic 0.
CED	Input	1	Active High, clock enable for the D port registers (DREG=1). If D port is not used, DREG should be set to 1 and CED tied to logic 0.
CEINMODE	Input	1	Active High, clock enable for the INMODE input registers (INMODEREG=1). If INMODE=0, CARRYINREG should be tied to logic 0.
CEM	Input	1	Active High, clock enable for the multiplier registers (MREG=1). If MREG=0, CEM should be tied to logic 0.
CEP	Input	1	Active High, clock enable for the output port registers (PREG=1). If PREG=0, PEM should be tied to logic 0.
CLK	Input	1	DSP slice clock input.
D[24:0]	Input	25	25-bit data input to the Pre-adder or alternative input to the Multiplier. Tie port to all ones if not used.
INMODE[4:0]	Input	5	Control input to select the arithmetic operation of the DSP slice in conjunction with ALUMODE and OPMODE. INMODE signals control the functionality of the signals and blocks that precede the Multiplier (including the pre-adder).
MULTSIGNIN	Input	1	Multiplier sign input from upstream cascaded DSP slice. Use for the purpose of sign extending the MACC output when greater than 48-bit output. Should only be connected to the MULTSIGNOUT output pin.
MULTSIGNOUT	Output	1	Multiplier sign output sent to downstream cascaded DSP slice. Use for the purpose of sign extending the MACC output when greater than 48-bit output. Should only be connected to the MULTISIGNIN input pin.
OPMODE[6:0]	Input	7	Control input to select the arithmetic operation of the DSP slice in conjunction with ALUMODE and INMODE.
OVERFLOW	Output	1	Active High output detects overflow in addition/accumulate if pattern detector is used and PREG=1.
P[47:0]	Output	48	Primary data output.
PATTERNBDETECT	Output	1	Active High pattern detection. Detects match of P and the bar of the selected PATTERN gated by the MASK. Result arrives on the same cycle as P.
PATTERNDETECT	Output	1	Active High pattern detection. Detects match of P and the selected PATTERN gated by the MASK. Result arrives on the same cycle as P.

Port	Direction	Width	Function
PCIN[47:0]	Input	48	Cascade input for Port P. If used, connect to PCOUT of upstream cascaded DSP slice. 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 DSP slice. If not used, leave unconnected.
RSTA	Input	1	Active High, synchronous reset for the A port registers (AREG=1 or 2). Tie to logic zero if not used.
RSTALLCARRYIN	Input	1	Active High, synchronous reset for all carry-in registers (CARRYINREG=1). Tie to logic zero if not used.
RSTALUMODE	Input	1	Active High, synchronous reset for the ALUMODE registers (ALUMODEREG=1). Tie to logic zero if not used.
RSTB	Input	1	Active High, synchronous reset for the B port registers (BREG=1 or 2). Tie to logic zero if not used.
RSTC	Input	1	Active High, synchronous reset for the C port registers (CREG=1). Tie to logic zero if not used.
RSTCTRL	Input	1	Active High, synchronous reset for the OPMODE and CARRYINSEL registers (OPMODEREG=1 and CARRYINSELREG=1). Tie to logic zero if not used.
RSTD	Input	1	Active High, synchronous reset for the D port registers (DREG=1). Tie to logic zero if not used.
RSTINMODE	Input	1	Active High, synchronous reset for the INMODE registers (INMODEREG=1). Tie to logic zero if not used.
RSTM	Input	1	Active High, synchronous reset for the multiplier registers (MREG=1). Tie to logic zero if not used.
RSTP	Input	1	Active High, synchronous reset for the output registers (PREG=1). Tie to logic zero if not used.
UNDERFLOW	Output	1	Active High output detects underflow in addition/accumulate if pattern detector is used and PREG = 1.

Design Entry Method

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

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
A_INPUT	String	"DIRECT", "CASCADE"	"DIRECT"	Selects between A and ACIN inputs.
ACASCREG	Integer	1, 0, 2	1	In conjunction with AREG, selects the number of A input registers on A cascade ACOUT. Must be equal to or one less than AREG value.
ADREG	Integer	1, 0	1	Selects usage of Pre-adder output (AD) Pipeline Registers. Set to 1 to use the AD Pipeline Registers.
ALUMODEREG	Integer	1, 0	1	Set to 1 to register the ALUMODE inputs.

Attribute	Data Type	Allowed Values	Default	Description
AREG	Integer	1, 0, 2	1	Selects number of pipeline stages for the A input.
AUTORESET_ PATDET	String	"NO_RESET", "RESET_MATCH", "RESET_NOT_ MATCH"	"NO_RESET"	Automatically reset DSP slice P Register (accumulated value or Counter Value) on the next clock cycle if pattern detect event has occurred on this clock cycle. The RESET_MATCH and RESET_NOT_MATCH settings distinguish between whether the DSP slice should cause auto reset of P Register on the next cycle if pattern is matched, or whenever pattern is not matched on the current cycle but was matched on the previous clock cycle.
B_INPUT	String	"DIRECT", "CASCADE"	"DIRECT"	Selects between B and BCIN inputs..
BCASCREG	Integer	1, 0, 2	1	In conjunction with BREG, selects the number of B input registers on B cascade BCOUT. Must be equal to or one less than BREG value.
BREG	Integer	1, 0, 2	1	Selects number of pipeline stages for the B input.
CARRYINREG	Integer	1, 0	1	Set to 1 to register the CARRYIN inputs.
CARRYINSELREG	Integer	1, 0	1	Set to 1 to register the CARRYINSEL inputs.
CREG	Integer	1, 0	1	Selects number of pipeline stages for the C input.
DREG	Integer	1, 0	1	Selects number of pipeline stages for the D input.
INMODEREG	Integer	1, 0	1	Set to 1 to register the INMODE inputs.
MASK	Hexa- decimal	48'h000000 000000 to 48'hffffffffffff	48'h3fff ffffffff	Mask to be used for pattern detector.
MREG	Integer	1, 0	1	Selects usage of multiplier output (M) pipeline registers. Set to 1 to use the M pipeline registers.
OPMODEREG	Integer	1, 0	1	Set to 1 to register the OPMODE inputs.
PATTERN	Hexa- decimal	48'h0000000 00000 to 48'hffffffffffff	All zeros	Pattern to be used for pattern detector.
PREG	Integer	1, 0	1	Set to 1 to register the P outputs. The registered outputs will include CARRYOUT, CARRYCASCOUT, MULTSIGNOUT, PATTERNB_DETECT, PATTERN_DETECT, and PCOUT.
SEL_MASK	String	"MASK", "C", "ROUNDING_ MODE1", "ROUNDING_ MODE2"	"MASK"	Selects mask to be used for pattern detector. The values C and MASK are for standard uses of the pattern detector (counter, overflow detection, etc.). ROUNDING_MODE1 (C-bar left shifted by 1) and ROUNDING_MODE2 (C-bar left shifted by 2) select special masks based on the optionally registered C port. These rounding modes can be used to implement convergent rounding in the DSP slice using the pattern detector as described in the *Virtex-6 FPGA DSP48E1 Block User Guide*.
SEL_PATTERN	String	"PATTERN", "C"	"PATTERN"	Selects pattern to be used for pattern detector.
USE_DPORT	Boolean	FALSE, TRUE	FALSE	Selects usage of the Pre-adder and D Port.

Attribute	Data Type	Allowed Values	Default	Description
USE_MULT	String	"MULTIPLY", "DYNAMIC", "NONE"	"MULTIPLY"	Selects usage of the Multiplier. Set to NONE to save power when using only the adder/Logic Unit. The DYNAMIC setting indicates that the user is switching between A*B and A:B operations on the fly and therefore needs to get the worst case timing of the two paths.
USE_PATTERN_ DETECT	String	"NO_PATDET", "PATDET"	"NO_PATDET"	Set to PATDET to enable pattern detection in the simulation model and speed files.
USE_SIMD	String	"ONE48", "FOUR12", "TWO24"	"ONE48"	Selects usage of the SIMD (Single Instruction Multiple Data) adder/Logic Unit. Selects between one 48-bit Logic Unit, two 24-bit Logic Units, or four 12-bit Logic Units. Note that all four 12 bit Logic Units share the same Instruction (i.e. all can subtract on the same cycle or add on the same cycle). This allows the 48 bit adder to be broken up into smaller adders for less computationally intensive applications. SIMD only has an effect on arithmetic operation (add, accumulate, subtract, etc.) and has no effect on logical operations.

VHDL Instantiation Template

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

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


-- DSP48E1: 48-bit Multi-Functional Arithmetic Block
--          Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

DSP48E1_inst : DSP48E1
generic map (
   -- Feature Control Attributes: Data Path Selection
   A_INPUT => "DIRECT",               -- Selects A input source, "DIRECT" (A port) or "CASCADE" (ACIN port)
   B_INPUT => "DIRECT",               -- Selects B input source, "DIRECT" (B port) or "CASCADE" (BCIN port)
   USE_DPORT => FALSE,                -- Select D port usage (TRUE or FALSE)
   USE_MULT => "MULTIPLY",            -- Select multiplier usage ("MULTIPLY", "DYNAMIC", or "NONE")
   -- Pattern Detector Attributes: Pattern Detection Configuration
   AUTORESET_PATDET => "NO_RESET",    -- "NO_RESET", "RESET_MATCH", "RESET_NOT_MATCH"
   MASK => X"3fffffffffff",           -- 48-bit mask value for pattern detect (1=ignore)
   PATTERN => X"000000000000",        -- 48-bit pattern match for pattern detect
   SEL_MASK => "MASK",                -- "C", "MASK", "ROUNDING_MODE1", "ROUNDING_MODE2"
   SEL_PATTERN => "PATTERN",          -- Select pattern value ("PATTERN" or "C")
   USE_PATTERN_DETECT => "NO_PATDET", -- Enable pattern detect ("PATDET" or "NO_PATDET")
   -- Register Control Attributes: Pipeline Register Configuration
   ACASCREG => 1,                     -- Number of pipeline stages between A/ACIN and ACOUT (0, 1 or 2)
   ADREG => 1,                        -- Number of pipeline stages for pre-adder (0 or 1)
   ALUMODEREG => 1,                   -- Number of pipeline stages for ALUMODE (0 or 1)
   AREG => 1,                         -- Number of pipeline stages for A (0, 1 or 2)
   BCASCREG => 1,                     -- Number of pipeline stages between B/BCIN and BCOUT (0, 1 or 2)
   BREG => 1,                         -- Number of pipeline stages for B (0, 1 or 2)
   CARRYINREG => 1,                   -- Number of pipeline stages for CARRYIN (0 or 1)
   CARRYINSELREG => 1,                -- Number of pipeline stages for CARRYINSEL (0 or 1)
   CREG => 1,                         -- Number of pipeline stages for C (0 or 1)
   DREG => 1,                         -- Number of pipeline stages for D (0 or 1)
   INMODEREG => 1,                    -- Number of pipeline stages for INMODE (0 or 1)
   MREG => 1,                         -- Number of multiplier pipeline stages (0 or 1)
   OPMODEREG => 1,                    -- Number of pipeline stages for OPMODE (0 or 1)
   PREG => 1,                         -- Number of pipeline stages for P (0 or 1)
   USE_SIMD => "ONE48"                -- SIMD selection ("ONE48", "TWO24", "FOUR12")
)
port map (
```

```
   -- Cascade: 30-bit (each) output: Cascade Ports
   ACOUT => ACOUT,                   -- 30-bit output: A port cascade output
   BCOUT => BCOUT,                   -- 18-bit output: B port cascade output
   CARRYCASCOUT => CARRYCASCOUT,     -- 1-bit output: Cascade carry output
   MULTSIGNOUT => MULTSIGNOUT,       -- 1-bit output: Multiplier sign cascade output
   PCOUT => PCOUT,                   -- 48-bit output: Cascade output
   -- Control: 1-bit (each) output: Control Inputs/Status Bits
   OVERFLOW => OVERFLOW,             -- 1-bit output: Overflow in add/acc output
   PATTERNBDETECT => PATTERNBDETECT, -- 1-bit output: Pattern bar detect output
   PATTERNDETECT => PATTERNDETECT,   -- 1-bit output: Pattern detect output
   UNDERFLOW => UNDERFLOW,           -- 1-bit output: Underflow in add/acc output
   -- Data: 4-bit (each) output: Data Ports
   CARRYOUT => CARRYOUT,             -- 4-bit output: Carry output
   P => P,                           -- 48-bit output: Primary data output
   -- Cascade: 30-bit (each) input: Cascade Ports
   ACIN => ACIN,                     -- 30-bit input: A cascade data input
   BCIN => BCIN,                     -- 18-bit input: B cascade input
   CARRYCASCIN => CARRYCASCIN,       -- 1-bit input: Cascade carry input
   MULTSIGNIN => MULTSIGNIN,         -- 1-bit input: Multiplier sign input
   PCIN => PCIN,                     -- 48-bit input: P cascade input
   -- Control: 4-bit (each) input: Control Inputs/Status Bits
   ALUMODE => ALUMODE,               -- 4-bit input: ALU control input
   CARRYINSEL => CARRYINSEL,         -- 3-bit input: Carry select input
   CEINMODE => CEINMODE,             -- 1-bit input: Clock enable input for INMODEREG
   CLK => CLK,                       -- 1-bit input: Clock input
   INMODE => INMODE,                 -- 5-bit input: INMODE control input
   OPMODE => OPMODE,                 -- 7-bit input: Operation mode input
   RSTINMODE => RSTINMODE,           -- 1-bit input: Reset input for INMODEREG
   -- Data: 30-bit (each) input: Data Ports
   A => A,                           -- 30-bit input: A data input
   B => B,                           -- 18-bit input: B data input
   C => C,                           -- 48-bit input: C data input
   CARRYIN => CARRYIN,               -- 1-bit input: Carry input signal
   D => D,                           -- 25-bit input: D data input
   -- Reset/Clock Enable: 1-bit (each) input: Reset/Clock Enable Inputs
   CEA1 => CEA1,                     -- 1-bit input: Clock enable input for 1st stage AREG
   CEA2 => CEA2,                     -- 1-bit input: Clock enable input for 2nd stage AREG
   CEAD => CEAD,                     -- 1-bit input: Clock enable input for ADREG
   CEALUMODE => CEALUMODE,           -- 1-bit input: Clock enable input for ALUMODERE
   CEB1 => CEB1,                     -- 1-bit input: Clock enable input for 1st stage BREG
   CEB2 => CEB2,                     -- 1-bit input: Clock enable input for 2nd stage BREG
   CEC => CEC,                       -- 1-bit input: Clock enable input for CREG
   CECARRYIN => CECARRYIN,           -- 1-bit input: Clock enable input for CARRYINREG
   CECTRL => CECTRL,                 -- 1-bit input: Clock enable input for OPMODEREG and CARRYINSELREG
   CED => CED,                       -- 1-bit input: Clock enable input for DREG
   CEM => CEM,                       -- 1-bit input: Clock enable input for MREG
   CEP => CEP,                       -- 1-bit input: Clock enable input for PREG
   RSTA => RSTA,                     -- 1-bit input: Reset input for AREG
   RSTALLCARRYIN => RSTALLCARRYIN,   -- 1-bit input: Reset input for CARRYINREG
   RSTALUMODE => RSTALUMODE,         -- 1-bit input: Reset input for ALUMODEREG
   RSTB => RSTB,                     -- 1-bit input: Reset input for BREG
   RSTC => RSTC,                     -- 1-bit input: Reset input for CREG
   RSTCTRL => RSTCTRL,               -- 1-bit input: Reset input for OPMODEREG and CARRYINSELREG
   RSTD => RSTD,                     -- 1-bit input: Reset input for DREG and ADREG
   RSTM => RSTM,                     -- 1-bit input: Reset input for MREG
   RSTP => RSTP                      -- 1-bit input: Reset input for PREG
);

-- End of DSP48E1_inst instantiation
```

Verilog Instantiation Template

```
// DSP48E1: 48-bit Multi-Functional Arithmetic Block
//          Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

DSP48E1 #(
   // Feature Control Attributes: Data Path Selection
   .A_INPUT("DIRECT"),               // Selects A input source, "DIRECT" (A port) or "CASCADE" (ACIN port)
   .B_INPUT("DIRECT"),               // Selects B input source, "DIRECT" (B port) or "CASCADE" (BCIN port)
   .USE_DPORT("FALSE"),              // Select D port usage (TRUE or FALSE)
```

```
   .USE_MULT("MULTIPLY"),            // Select multiplier usage ("MULTIPLY", "DYNAMIC", or "NONE")
   // Pattern Detector Attributes: Pattern Detection Configuration
   .AUTORESET_PATDET("NO_RESET"),    // "NO_RESET", "RESET_MATCH", "RESET_NOT_MATCH"
   .MASK(48'h3fffffffffff),          // 48-bit mask value for pattern detect (1=ignore)
   .PATTERN(48'h000000000000),       // 48-bit pattern match for pattern detect
   .SEL_MASK("MASK"),                // "C", "MASK", "ROUNDING_MODE1", "ROUNDING_MODE2"
   .SEL_PATTERN("PATTERN"),          // Select pattern value ("PATTERN" or "C")
   .USE_PATTERN_DETECT("NO_PATDET"), // Enable pattern detect ("PATDET" or "NO_PATDET")
   // Register Control Attributes: Pipeline Register Configuration
   .ACASCREG(1),                     // Number of pipeline stages between A/ACIN and ACOUT (0, 1 or 2)
   .ADREG(1),                        // Number of pipeline stages for pre-adder (0 or 1)
   .ALUMODEREG(1),                   // Number of pipeline stages for ALUMODE (0 or 1)
   .AREG(1),                         // Number of pipeline stages for A (0, 1 or 2)
   .BCASCREG(1),                     // Number of pipeline stages between B/BCIN and BCOUT (0, 1 or 2)
   .BREG(1),                         // Number of pipeline stages for B (0, 1 or 2)
   .CARRYINREG(1),                   // Number of pipeline stages for CARRYIN (0 or 1)
   .CARRYINSELREG(1),                // Number of pipeline stages for CARRYINSEL (0 or 1)
   .CREG(1),                         // Number of pipeline stages for C (0 or 1)
   .DREG(1),                         // Number of pipeline stages for D (0 or 1)
   .INMODEREG(1),                    // Number of pipeline stages for INMODE (0 or 1)
   .MREG(1),                         // Number of multiplier pipeline stages (0 or 1)
   .OPMODEREG(1),                    // Number of pipeline stages for OPMODE (0 or 1)
   .PREG(1),                         // Number of pipeline stages for P (0 or 1)
   .USE_SIMD("ONE48")                // SIMD selection ("ONE48", "TWO24", "FOUR12")
)
DSP48E1_inst (
   // Cascade: 30-bit (each) output: Cascade Ports
   .ACOUT(ACOUT),                   // 30-bit output: A port cascade output
   .BCOUT(BCOUT),                   // 18-bit output: B port cascade output
   .CARRYCASCOUT(CARRYCASCOUT),     // 1-bit output: Cascade carry output
   .MULTSIGNOUT(MULTSIGNOUT),       // 1-bit output: Multiplier sign cascade output
   .PCOUT(PCOUT),                   // 48-bit output: Cascade output
   // Control: 1-bit (each) output: Control Inputs/Status Bits
   .OVERFLOW(OVERFLOW),             // 1-bit output: Overflow in add/acc output
   .PATTERNBDETECT(PATTERNBDETECT), // 1-bit output: Pattern bar detect output
   .PATTERNDETECT(PATTERNDETECT),   // 1-bit output: Pattern detect output
   .UNDERFLOW(UNDERFLOW),           // 1-bit output: Underflow in add/acc output
   // Data: 4-bit (each) output: Data Ports
   .CARRYOUT(CARRYOUT),             // 4-bit output: Carry output
   .P(P),                           // 48-bit output: Primary data output
   // Cascade: 30-bit (each) input: Cascade Ports
   .ACIN(ACIN),                     // 30-bit input: A cascade data input
   .BCIN(BCIN),                     // 18-bit input: B cascade input
   .CARRYCASCIN(CARRYCASCIN),       // 1-bit input: Cascade carry input
   .MULTSIGNIN(MULTSIGNIN),         // 1-bit input: Multiplier sign input
   .PCIN(PCIN),                     // 48-bit input: P cascade input
   // Control: 4-bit (each) input: Control Inputs/Status Bits
   .ALUMODE(ALUMODE),               // 4-bit input: ALU control input
   .CARRYINSEL(CARRYINSEL),         // 3-bit input: Carry select input
   .CEINMODE(CEINMODE),             // 1-bit input: Clock enable input for INMODEREG
   .CLK(CLK),                       // 1-bit input: Clock input
   .INMODE(INMODE),                 // 5-bit input: INMODE control input
   .OPMODE(OPMODE),                 // 7-bit input: Operation mode input
   .RSTINMODE(RSTINMODE),           // 1-bit input: Reset input for INMODEREG
   // Data: 30-bit (each) input: Data Ports
   .A(A),                           // 30-bit input: A data input
   .B(B),                           // 18-bit input: B data input
   .C(C),                           // 48-bit input: C data input
   .CARRYIN(CARRYIN),               // 1-bit input: Carry input signal
   .D(D),                           // 25-bit input: D data input
   // Reset/Clock Enable: 1-bit (each) input: Reset/Clock Enable Inputs
   .CEA1(CEA1),                     // 1-bit input: Clock enable input for 1st stage AREG
   .CEA2(CEA2),                     // 1-bit input: Clock enable input for 2nd stage AREG
   .CEAD(CEAD),                     // 1-bit input: Clock enable input for ADREG
   .CEALUMODE(CEALUMODE),           // 1-bit input: Clock enable input for ALUMODERE
   .CEB1(CEB1),                     // 1-bit input: Clock enable input for 1st stage BREG
   .CEB2(CEB2),                     // 1-bit input: Clock enable input for 2nd stage BREG
   .CEC(CEC),                       // 1-bit input: Clock enable input for CREG
   .CECARRYIN(CECARRYIN),           // 1-bit input: Clock enable input for CARRYINREG
   .CECTRL(CECTRL),                 // 1-bit input: Clock enable input for OPMODEREG and CARRYINSELREG
   .CED(CED),                       // 1-bit input: Clock enable input for DREG
   .CEM(CEM),                       // 1-bit input: Clock enable input for MREG
   .CEP(CEP),                       // 1-bit input: Clock enable input for PREG
```

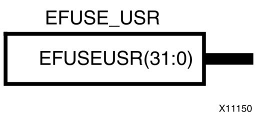
```
   .RSTA(RSTA),                     // 1-bit input: Reset input for AREG
   .RSTALLCARRYIN(RSTALLCARRYIN),   // 1-bit input: Reset input for CARRYINREG
   .RSTALUMODE(RSTALUMODE),         // 1-bit input: Reset input for ALUMODEREG
   .RSTB(RSTB),                     // 1-bit input: Reset input for BREG
   .RSTC(RSTC),                     // 1-bit input: Reset input for CREG
   .RSTCTRL(RSTCTRL),               // 1-bit input: Reset input for OPMODEREG and CARRYINSELREG
   .RSTD(RSTD),                     // 1-bit input: Reset input for DREG and ADREG
   .RSTM(RSTM),                     // 1-bit input: Reset input for MREG
   .RSTP(RSTP)                      // 1-bit input: Reset input for PREG
);

// End of DSP48E1_inst instantiation
```

For More Information

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

EFUSE_USR

Primitive: 32-bit non-volatile design ID



Introduction

Provides internal access via JTAG to the 32 non-volatile fuses that can store bits specific to the design (e.g., a unique ID associated with each design).

Port Descriptions

Port	Direction	Width	Function
EFUSEUSR[31:0]	Output	32	User E-Fuse register value

Design Entry Method

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

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
SIM_EFUSE_VALUE	Hexadecimal	32'h00000000 to 32'hffffffff	32'h00000000	Value of the 32-bit non-volatile design ID used in 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;


-- EFUSE_USR: 32-bit non-volatile design ID
--            Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

EFUSE_USR_inst : EFUSE_USR
generic map (
   SIM_EFUSE_VALUE => X"00000000"  -- Value of the 32-bit non-volatile design ID used in simulation
)
port map (
   EFUSEUSR => EFUSEUSR  -- 32-bit output: User E-Fuse register value output
);

-- End of EFUSE_USR_inst instantiation
```

Verilog Instantiation Template

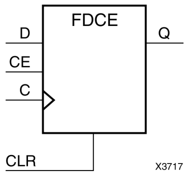
```
// EFUSE_USR: 32-bit non-volatile design ID
//            Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

EFUSE_USR #(
   .SIM_EFUSE_VALUE(32'h00000000)  // Value of the 32-bit non-volatile design ID used in simulation
)
EFUSE_USR_inst (
   .EFUSEUSR(EFUSEUSR)  // 32-bit output: User E-Fuse register value output
);

// End of EFUSE_USR_inst instantiation
```

For More Information

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

FDCE

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



Introduction

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

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

VHDL 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).
--       Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

-- End of FDCE_inst instantiation
```

Verilog Instantiation Template

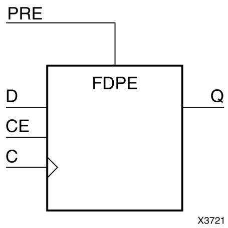
```
// FDCE: Single Data Rate D Flip-Flop with Asynchronous Clear and
//       Clock Enable (posedge clk).
//       Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

FDCE #(
   .INIT(1'b0) // Initial value of register (1'b0 or 1'b1)
) 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 *Virtex-6 FPGA User Documentation (User Guides and Data Sheets)*.

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

VHDL 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).
--       Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

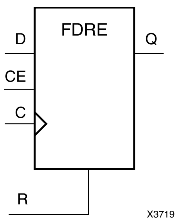
```
// FDPE: Single Data Rate D Flip-Flop with Asynchronous Preset and
//       Clock Enable (posedge clk).
//       Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

FDPE #(
   .INIT(1'b0) // Initial value of register (1'b0 or 1'b1)
) 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 *Virtex-6 FPGA User Documentation (User Guides and Data Sheets)*.

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

VHDL 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).
--       Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

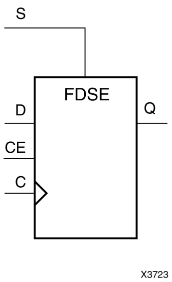
```
// FDRE: Single Data Rate D Flip-Flop with Synchronous Reset and
//       Clock Enable (posedge clk).
//       Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

FDRE #(
   .INIT(1'b0) // Initial value of register (1'b0 or 1'b1)
) 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 *Virtex-6 FPGA User Documentation (User Guides and Data Sheets)*.

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

VHDL 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).
--       Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

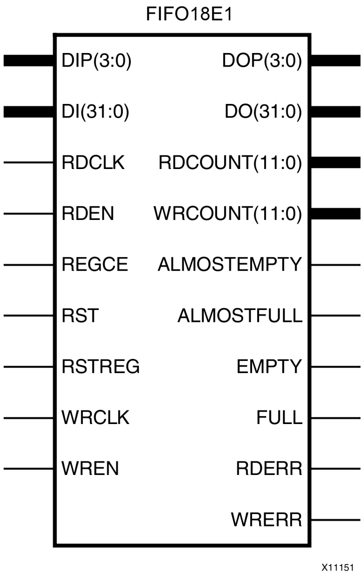
```
// FDSE: Single Data Rate D Flip-Flop with Synchronous Set and
//       Clock Enable (posedge clk).
//       Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

FDSE #(
   .INIT(1'b0) // Initial value of register (1'b0 or 1'b1)
) 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 *Virtex-6 FPGA User Documentation (User Guides and Data Sheets)*.

FIFO18E1

Primitive: 18 k-bit FIFO (First In, First Out) Block RAM Memory



Introduction

Virtex®-6 devices contain several block RAM memories, each of which can be separately configured as a FIFO, an automatic error-correction RAM, or as a general-purpose 36 kb or 18 kb RAM/ROM memory. These Block RAM memories offer fast and flexible storage of large amounts of on-chip data. The FIFO18E1 uses the FIFO control logic and the 18 kb block RAM. This primitive can be used in a 4-bit wide by 4K deep, 9-bit wide by 2K deep, 18-bit wide by 1K deep, or a 36-bit wide by 512 deep configuration. The primitive can be configured in either synchronous or dual-clock (asynchronous) mode, with all associated FIFO flags and status signals When using the dual-clock mode with independent clocks, depending on the offset between read and write clock edges, the Empty, Almost Empty, Full and Almost Full flags can deassert one cycle later. Due to the asynchronous nature of the clocks the simulation model only reflects the deassertion latency cycles listed in the User Guide.

Note For a 36-bit wide by 512 deep FIFO, the FIFO18_36 mode must be used. For deeper or wider configurations of the FIFO, the FIFO36E1 can be used. If error-correction circuitry is desired, the FIFO36E1 with FIFO36_72 mode must be used.

Port Descriptions

Port	Direction	Width	Function
ALMOSTEMPTY	Output	1	Programmable flag to indicate the FIFO is almost empty. ALMOST_EMPTY_OFFSET attribute specifies the threshold where this flag is triggered relative to full/empty.
ALMOSTFULL	Output	1	Programmable flag to indicate that the FIFO is almost full. The ALMOST_FULL_OFFSET attribute specifies the threshold where this flag is triggered relative to full/empty.
DI[31:0]	Input	32	FIFO data input bus.
DIP[3:0]	Input	4	FIFO parity data input bus.
DO[31:0]	Output	32	FIFO data output bus.

Port	Direction	Width	Function
DOP[3:0]	Output	4	FIFO parity data output bus.
EMPTY	Output	1	Active High logic to indicate that the FIFO is currently empty.
FULL	Output	1	Active High logic indicates that the FIFO is full.
RDEN	Input	1	Active High FIFO read enable.
REGCE	Input	1	Output register clock enable for pipelined synchronous FIFO.
RST	Input	1	Active High (FIFO logic) asynchronous reset (for dual-clock FIFO), synchronous reset (synchronous FIFO) for 3 CLK cycles.
RSTREG	Input	1	Output register synchronous set/reset.
WRCLK, RDCLK	Input	1	FIFO read and write clocks (positive edge triggered).
WRCOUNT, RDCOUNT	Output	12	FIFO write/read pointer.
WREN	Input	1	Active High FIFO write enable.
WRERR, RDERR	Output	1	• WRERR indicates that a write occurred while the FIFO was full. • RDERR indicates that a read occurred while the FIFO was empty.

Design Entry Method

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

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
ALMOST_EMPTY_ OFFSET	Hexa- decimal	13'h0000 to 13'h8191	13'h0080	Specifies the amount of data contents in the RAM to trigger the ALMOST_EMPTY flag.
ALMOST_FULL_ OFFSET	Hexa- decimal	13'h0000 to 13'h8191	13'h0080	Specifies the amount of data contents in the RAM to trigger the ALMOST_FULL flag.
DATA_WIDTH	Integer	4, 9, 18, 36	4	Specifies the desired data width for the FIFO.
DO_REG	Integer	1, 0	1	Data pipeline register for EN_SYN.
EN_SYN	Boolean	FALSE, TRUE	FALSE	Specifies whether the FIFO is operating in either dual-clock (two independent clocks) or synchronous (single clock) mode. Dual-clock must use DO_REG=1.
FIFO_MODE	String	"FIFO18", "FIFO18_36"	"FIFO18"	Selects FIFO18 or FIFO18_36 mode.

Attribute	Data Type	Allowed Values	Default	Description
FIRST_WORD_FALL_ THROUGH	Boolean	FALSE, TRUE	FALSE	If TRUE, the first write to the FIFO appears on DO without a first RDEN assertion.
INIT	Hexa- decimal	Any 36 bit Value	All zeros	Specifies the initial value on the DO output after configuration.
SRVAL	Hexa- decimal	Any 36 bit Value	All zeros	Specifies the output value of the FIFO upon assertion of the synchronous reset (RSTREG) signal. Only valid for DO_REG=1.

VHDL Instantiation Template

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

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


-- FIFO18E1: 18KB FIFO (First In, First Out) Block RAM Memory
--           Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

FIFO18E1_inst : FIFO18E1
generic map (
   ALMOST_EMPTY_OFFSET => X"0080",   -- Sets the almost empty threshold
   ALMOST_FULL_OFFSET => X"0080",    -- Sets almost full threshold
   DATA_WIDTH => 4,                  -- Sets data width to 4, 9, 18, or 36
   DO_REG => 1,                      -- Enable output register (0 or 1) Must be 1 if EN_SYN = FALSE
   EN_SYN => FALSE,                  -- Specifies FIFO as dual-clock (FALSE) or Synchronous (TRUE)
   FIFO_MODE => "FIFO18",            -- Sets mode to FIFO18 or FIFO18_36
   FIRST_WORD_FALL_THROUGH => FALSE, -- Sets the FIFO FWFT to TRUE or FALSE
   INIT => X"000000000",             -- Initial values on output port
   SRVAL => X"000000000"             -- Set/Reset value for output port
)
port map (
   -- Read Data: 32-bit (each) output: Read output data
   DO => DO,                   -- 32-bit output: data output
   DOP => DOP,                 -- 4-bit output: parity data output
   -- Status: 1-bit (each) output: Flags and other FIFO status outputs
   ALMOSTEMPTY => ALMOSTEMPTY, -- 1-bit output: almost empty output flag
   ALMOSTFULL => ALMOSTFULL,   -- 1-bit output: almost full output flag
   EMPTY => EMPTY,             -- 1-bit output: empty output flag
   FULL => FULL,               -- 1-bit output: full output flag
   RDCOUNT => RDCOUNT,         -- 12-bit output: read count output
   RDERR => RDERR,             -- 1-bit output: read error output
   WRCOUNT => WRCOUNT,         -- 12-bit output: write count output
   WRERR => WRERR,             -- 1-bit output: write error
   -- Read Control Signals: 1-bit (each) input: Read clock, enable and reset input signals
   RDCLK => RDCLK,             -- 1-bit input: read clock input
   RDEN => RDEN,               -- 1-bit input: read enable input
   REGCE => REGCE,             -- 1-bit input: clock enable input
   RST => RST,                 -- 1-bit input: reset input
   RSTREG => RSTREG,           -- 1-bit input: output register set/reset
   -- Write Control Signals: 1-bit (each) input: Write clock and enable input signals
   WRCLK => WRCLK,             -- 1-bit input: write clock input
   WREN => WREN,               -- 1-bit input: write enable input
   -- Write Data: 32-bit (each) input: Write input data
   DI => DI,                   -- 32-bit input: data input
   DIP => DIP                  -- 4-bit input: parity input
);

-- End of FIFO18E1_inst instantiation
```

Verilog Instantiation Template

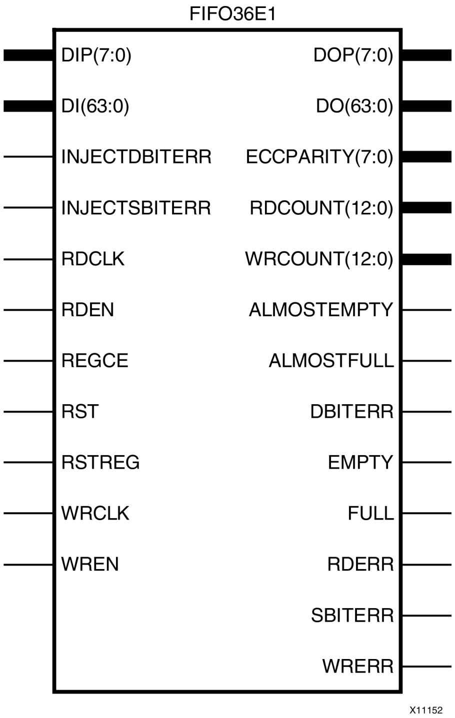
```
// FIFO18E1: 18KB FIFO (First In, First Out) Block RAM Memory
//           Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

FIFO18E1 #(
   .ALMOST_EMPTY_OFFSET(13'h0080),    // Sets the almost empty threshold
   .ALMOST_FULL_OFFSET(13'h0080),     // Sets almost full threshold
   .DATA_WIDTH(4),                    // Sets data width to 4, 9, 18, or 36
   .DO_REG(1),                        // Enable output register (0 or 1) Must be 1 if EN_SYN = FALSE
   .EN_SYN("FALSE"),                  // Specifies FIFO as dual-clock (FALSE) or Synchronous (TRUE)
   .FIFO_MODE("FIFO18"),              // Sets mode to FIFO18 or FIFO18_36
   .FIRST_WORD_FALL_THROUGH("FALSE"), // Sets the FIFO FWFT to TRUE or FALSE
   .INIT(36'h000000000),              // Initial values on output port
   .SRVAL(36'h000000000)              // Set/Reset value for output port
)
FIFO18E1_inst (
   // Read Data: 32-bit (each) output: Read output data
   .DO(DO),                   // 32-bit output: data output
   .DOP(DOP),                 // 4-bit output: parity data output
   // Status: 1-bit (each) output: Flags and other FIFO status outputs
   .ALMOSTEMPTY(ALMOSTEMPTY), // 1-bit output: almost empty output flag
   .ALMOSTFULL(ALMOSTFULL),   // 1-bit output: almost full output flag
   .EMPTY(EMPTY),             // 1-bit output: empty output flag
   .FULL(FULL),               // 1-bit output: full output flag
   .RDCOUNT(RDCOUNT),         // 12-bit output: read count output
   .RDERR(RDERR),             // 1-bit output: read error output
   .WRCOUNT(WRCOUNT),         // 12-bit output: write count output
   .WRERR(WRERR),             // 1-bit output: write error
   // Read Control Signals: 1-bit (each) input: Read clock, enable and reset input signals
   .RDCLK(RDCLK),             // 1-bit input: read clock input
   .RDEN(RDEN),               // 1-bit input: read enable input
   .REGCE(REGCE),             // 1-bit input: clock enable input
   .RST(RST),                 // 1-bit input: reset input
   .RSTREG(RSTREG),           // 1-bit input: output register set/reset
   // Write Control Signals: 1-bit (each) input: Write clock and enable input signals
   .WRCLK(WRCLK),             // 1-bit input: write clock input
   .WREN(WREN),               // 1-bit input: write enable input
   // Write Data: 32-bit (each) input: Write input data
   .DI(DI),                   // 32-bit input: data input
   .DIP(DIP)                  // 4-bit input: parity input
);

// End of FIFO18E1_inst instantiation
```

For More Information

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

FIFO36E1

Primitive: 36 kb FIFO (First In, First Out) Block RAM Memory



Introduction

Virtex®-6 devices contain several block RAM memories that can be configured as FIFOs, automatic error-correction RAM, or general-purpose 36 kb or 18 kb RAM/ROM memories. These block RAM memories offer fast and flexible storage of large amounts of on-chip data. The FIFO36E1 allows access to the block RAM in the 36 kb FIFO configurations. This component can be configured and used as a 4-bit wide by 8K deep, 9-bit by 4K deep, 18-bit by 2K deep, 36-bit wide by 1K deep, or 72-bit wide by 512 deep synchronous or dual-clock (asynchronous) FIFO RAM with all associated FIFO flags.

When using the dual-clock mode with independent clocks, depending on the offset between read and write clock edges, the Empty, Almost Empty, Full, and Almost Full flags can deassert one cycle later. Due to the asynchronous nature of the clocks, the simulation model only reflects the deassertion latency cycles listed in the User Guide.

Note For a 72-bit wide by 512 deep FIFO, the FIFO36_72 mode must be used. For smaller configurations of the FIFO, the FIFO18E1 can be used. If error-correction circuitry is desired, the FIFO36_72 mode must be used.

Port Descriptions

Port	Direction	Width	Function
ALMOSTEMPTY	Output	1	Programmable flag to indicate the FIFO is almost empty. ALMOST_EMPTY_OFFSET attribute specifies where to trigger this flag.
ALMOSTFULL	Output	1	Programmable flag to indicate the FIFO is almost full. ALMOST_FULL_OFFSET attribute specifies where to trigger this flag.
DBITERR	Output	1	Status output from ECC function to indicate a double bit error was detected. EN_ECC_READ needs to be TRUE in order to use this functionality.

Port	Direction	Width	Function
DI[63:0]	Input	64	FIFO data input bus.
DIP[7:0]	Input	8	FIFO parity data input bus.
DO[63:0]	Output	64	FIFO data output bus.
DOP[7:0]	Output	8	FIFO parity data output bus.
ECCPARITY[7:0]	Output	8	8-bit data generated by the ECC encoder used by the ECC decoder for memory error detection and correction.
EMPTY	Output	1	Active high logic to indicate that the FIFO is currently empty.
FULL	Output	1	Active high logic indicates that the FIFO is full.
INJECTDBITERR	Input	1	Inject a double bit error if ECC feature is used.
INJECTSBITERR	Input	1	Inject a single bit error if ECC feature is used.
RDEN	Input	1	Active high FIFO read enable.
REGCE	Input	1	Output register clock enable for pipelined synchronous FIFO.
RST	Input	1	Active high (FIFO logic) asynchronous reset (for dual-clock FIFO), synchronous reset (synchronous FIFO) for 3 CLK cycles.
RSTREG	Input	1	Output register synchronous set/reset.
SBITERR	Output	1	Status output from ECC function to indicate a single bit error was detected. EN_ECC_READ needs to be TRUE in order to use this functionality.
WRCLK, RDCLK	Input	1	FIFO read and write clocks (positive edge triggered).
WRCOUNT, RDCOUNT	Output	13	FIFO write/read pointer.
WREN	Input	1	Active high FIFO write enable.
WRERR, RDERR	Output	1	• WRERR indicates that a write occurred while the FIFO was full. • RDERR indicates that a read occurred while the FIFO was empty.

Design Entry Method

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

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
ALMOST_EMPTY_ OFFSET	Hexa- decimal	13'h0000 to 13'h8191	13'h0080	Specifies the amount of data contents in the RAM to trigger the ALMOST_EMPTY flag.
ALMOST_FULL_ OFFSET	Hexa- decimal	13'h0000 to 13'h8191	13'h0080	Specifies the amount of data contents in the RAM to trigger the ALMOST_FULL flag.
DATA_WIDTH	Integer	4, 9, 18, 36, 72	4	Specifies the desired data width for the FIFO.
DO_REG	Integer	1, 0	1	Enable output register to the FIFO for improved clock-to-out timing at the expense of added read latency (one pipeline delay). DO_REG must be 1 when EN_SYN is set to FALSE.
EN_ECC_READ	Boolean	FALSE, TRUE	FALSE	Enable the ECC decoder circuitry.
EN_ECC_WRITE	Boolean	FALSE, TRUE	FALSE	Enable the ECC encoder circuitry.
EN_SYN	Boolean	FALSE, TRUE	FALSE	When FALSE, specifies the FIFO to be used in asynchronous mode (two independent clock). When TRUE in synchronous (a single clock) operation.
FIFO_MODE	String	"FIFO36, "FIFO36_72"	"FIFO36"	Selects FIFO36 or FIFO36_72 mode.
FIRST_WORD_FALL_ THROUGH	Boolean	FALSE, TRUE	FALSE	If TRUE, the first write to the FIFO will appear on DO without an RDEN assertion.
INIT	Hexa- decimal	Any 72 bit Value	All zeros	Specifies the initial value on the DO output after configuration.
SRVAL	Hexa- decimal	Any 72 bit Value	All zeros	Specifies the output value of the FIFO upon assertion of the synchronous reset (RSTREG) signal. Only valid for DO_REG=1.

VHDL Instantiation Template

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

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


-- FIFO36E1: 36KB FIFO (First In, First Out) Block RAM Memory
--           Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

FIFO36E1_inst : FIFO36E1
generic map (
   ALMOST_EMPTY_OFFSET => X"0080",    -- Sets the almost empty threshold
   ALMOST_FULL_OFFSET => X"0080",     -- Sets almost full threshold
   DATA_WIDTH => 4,                   -- Sets data width to 4, 9, 18, 36, or 72
   DO_REG => 1,                       -- Enable output register (0 or 1) Must be 1 if EN_SYN = FALSE
   EN_ECC_READ => FALSE,              -- Enable ECC decoder, TRUE or FALSE
   EN_ECC_WRITE => FALSE,             -- Enable ECC encoder, TRUE or FALSE
   EN_SYN => FALSE,                   -- Specifies FIFO as Asynchronous (FALSE) or Synchronous (TRUE)
   FIFO_MODE => "FIFO36",             -- Sets mode to FIFO36 or FIFO36_72
   FIRST_WORD_FALL_THROUGH => FALSE,  -- Sets the FIFO FWFT to TRUE or FALSE
   INIT => X"000000000000000000",     -- Initial values on output port
   SRVAL => X"000000000000000000"     -- Set/Reset value for output port
```

```
)
port map (
   -- ECC Signals: 1-bit (each) output: Error Correction Circuitry ports
   DBITERR => DBITERR,             -- 1-bit output: double bit error status output
   ECCPARITY => ECCPARITY,         -- 8-bit output: generated error correction parity
   SBITERR => SBITERR,             -- 1-bit output: single bit error status output
   -- Read Data: 64-bit (each) output: Read output data
   DO => DO,                       -- 64-bit output: data output
   DOP => DOP,                     -- 8-bit output: parity data output
   -- Status: 1-bit (each) output: Flags and other FIFO status outputs
   ALMOSTEMPTY => ALMOSTEMPTY,     -- 1-bit output: almost empty output flag
   ALMOSTFULL => ALMOSTFULL,       -- 1-bit output: almost full output flag
   EMPTY => EMPTY,                 -- 1-bit output: empty output flag
   FULL => FULL,                   -- 1-bit output: full output flag
   RDCOUNT => RDCOUNT,             -- 13-bit output: read count output
   RDERR => RDERR,                 -- 1-bit output: read error output
   WRCOUNT => WRCOUNT,             -- 13-bit output: write count output
   WRERR => WRERR,                 -- 1-bit output: write error
   -- ECC Signals: 1-bit (each) input: Error Correction Circuitry ports
   INJECTDBITERR => INJECTDBITERR, -- 1-bit input: Inject a double bit error
   INJECTSBITERR => INJECTSBITERR,
   -- Read Control Signals: 1-bit (each) input: Read clock, enable and reset input signals
   RDCLK => RDCLK,                 -- 1-bit input: read clock input
   RDEN => RDEN,                   -- 1-bit input: read enable input
   REGCE => REGCE,                 -- 1-bit input: clock enable input
   RST => RST,                     -- 1-bit input: reset input
   RSTREG => RSTREG,               -- 1-bit input: output register set/reset
   -- Write Control Signals: 1-bit (each) input: Write clock and enable input signals
   WRCLK => WRCLK,                 -- 1-bit input: write clock input
   WREN => WREN,                   -- 1-bit input: write enable input
   -- Write Data: 64-bit (each) input: Write input data
   DI => DI,                       -- 64-bit input: data input
   DIP => DIP                      -- 8-bit input: parity input
);

-- End of FIFO36E1_inst instantiation
```

Verilog Instantiation Template

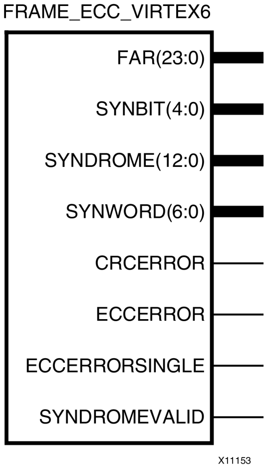
```
// FIFO36E1: 36KB FIFO (First In, First Out) Block RAM Memory
//           Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

FIFO36E1 #(
   .ALMOST_EMPTY_OFFSET(13'h0080),    // Sets the almost empty threshold
   .ALMOST_FULL_OFFSET(13'h0080),     // Sets almost full threshold
   .DATA_WIDTH(4),                    // Sets data width to 4, 9, 18, 36, or 72
   .DO_REG(1),                        // Enable output register (0 or 1) Must be 1 if EN_SYN = FALSE
   .EN_ECC_READ("FALSE"),             // Enable ECC decoder, TRUE or FALSE
   .EN_ECC_WRITE("FALSE"),            // Enable ECC encoder, TRUE or FALSE
   .EN_SYN("FALSE"),                  // Specifies FIFO as Asynchronous (FALSE) or Synchronous (TRUE)
   .FIFO_MODE("FIFO36"),              // Sets mode to FIFO36 or FIFO36_72
   .FIRST_WORD_FALL_THROUGH("FALSE"), // Sets the FIFO FWFT to TRUE or FALSE
   .INIT(72'h000000000000000000),     // Initial values on output port
   .SRVAL(72'h000000000000000000)     // Set/Reset value for output port
)
FIFO36E1_inst (
   // ECC Signals: 1-bit (each) output: Error Correction Circuitry ports
   .DBITERR(DBITERR),             // 1-bit output: double bit error status output
   .ECCPARITY(ECCPARITY),         // 8-bit output: generated error correction parity
   .SBITERR(SBITERR),             // 1-bit output: single bit error status output
   // Read Data: 64-bit (each) output: Read output data
   .DO(DO),                       // 64-bit output: data output
   .DOP(DOP),                     // 8-bit output: parity data output
   // Status: 1-bit (each) output: Flags and other FIFO status outputs
   .ALMOSTEMPTY(ALMOSTEMPTY),     // 1-bit output: almost empty output flag
   .ALMOSTFULL(ALMOSTFULL),       // 1-bit output: almost full output flag
   .EMPTY(EMPTY),                 // 1-bit output: empty output flag
   .FULL(FULL),                   // 1-bit output: full output flag
   .RDCOUNT(RDCOUNT),             // 13-bit output: read count output
   .RDERR(RDERR),                 // 1-bit output: read error output
   .WRCOUNT(WRCOUNT),             // 13-bit output: write count output
   .WRERR(WRERR),                 // 1-bit output: write error
   // ECC Signals: 1-bit (each) input: Error Correction Circuitry ports
   .INJECTDBITERR(INJECTDBITERR), // 1-bit input: Inject a double bit error
   .INJECTSBITERR(INJECTSBITERR),
   // Read Control Signals: 1-bit (each) input: Read clock, enable and reset input signals
   .RDCLK(RDCLK),                 // 1-bit input: read clock input
   .RDEN(RDEN),                   // 1-bit input: read enable input
   .REGCE(REGCE),                 // 1-bit input: clock enable input
   .RST(RST),                     // 1-bit input: reset input
   .RSTREG(RSTREG),               // 1-bit input: output register set/reset
   // Write Control Signals: 1-bit (each) input: Write clock and enable input signals
   .WRCLK(WRCLK),                 // 1-bit input: write clock input
   .WREN(WREN),                   // 1-bit input: write enable input
   // Write Data: 64-bit (each) input: Write input data
   .DI(DI),                       // 64-bit input: data input
   .DIP(DIP)                      // 8-bit input: parity input
);

// End of FIFO36E1_inst instantiation
```

For More Information

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

FRAME_ECC_VIRTEX6

Primitive: Virtex®-6 Configuration Frame Error Detection and Correction Circuitry



Introduction

This design element enables the dedicated, built-in ECC (Error Detection and Correction Circuitry) for the configuration memory of the FPGA. This element contains outputs that allow monitoring of the status of the ECC circuitry and the status of the readback CRC circuitry.

SEU Correction feature provides hardware version to allow automatic correction of single-bit errors. New additional outputs used by the correction feature include the decoding of the Hamming code syndrome for use by the soft core.

Port Descriptions

Port	Direction	Width	Function
CRCERROR	Output	1	Output indicating a CRC error.
ECCERROR	Output	1	Output indicating a ECC error.
ECCERRORSINGLE	Output	1	Indicates single-bit Frame ECC error detected.
FAR[23:0]	Output	24	Frame Address Register Value.
SYNBIT[4:0]	Output	5	Bit address of error.
SYNDROME[12:0]	Output	13	Output location of erroneous bit
SYNDROMEVALID	Output	1	Frame ECC output indicating the SYNDROME output is valid.
SYNWORD[6:0]	Output	7	Word in the frame where an ECC error has been detected.

Design Entry Method

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

Available Attributes

Attribute	Data Type	Allowed_Values	Default	Description
FARSRC	String	"EFAR", "FAR"	"EFAR"	EFAR Determines if the output of FAR[23:0] configuration register points to the FAR or EFAR. Sets configuration option register bit CTL0[7].
FRAME_RBT_IN_ FILENAME	String	String representing file name and location	None	This file is output by the ICAP_VIRTEX6 model and it contains Frame Data information for the Raw Bitstream (RBT) file. The FRAME_ECC model will parse this file, calculate ECC and output any error conditions.

VHDL Instantiation Template

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

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


-- FRAME_ECC_VIRTEX6: Configuration Frame Error Correction
--                    Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

FRAME_ECC_VIRTEX6_inst : FRAME_ECC_VIRTEX6
generic map (
   FARSRC => "EFAR",                 -- Determines if the output of FAR[23:0] configuration register points
                                     -- to the FAR or EFAR. Sets configuration option register bit CTL0[7].
   FRAME_RBT_IN_FILENAME => "NONE"   -- This file is output by the ICAP_VIRTEX6 model and it contains Frame
                                     -- Data information for the Raw Bitstream (RBT) file. The FRAME_ECC
                                     -- model will parse this file, calculate ECC and output any error
                                     -- conditions.
)
port map (
   CRCERROR => CRCERROR,              -- 1-bit output: Output indicating a CRC error
   ECCERROR => ECCERROR,              -- 1-bit output: Output indicating an ECC error
   ECCERRORSINGLE => ECCERRORSINGLE,  -- 1-bit output: Output Indicating single-bit Frame ECC error detected.
   FAR => FAR,                        -- 24-bit output: Frame Address Register Value output
   SYNBIT => SYNBIT,                  -- 5-bit output: Output bit address of error
   SYNDROME => SYNDROME,              -- 13-bit output: Output location of erroneous bit
   SYNDROMEVALID => SYNDROMEVALID,    -- 1-bit output: Frame ECC output indicating the SYNDROME output is
                                      -- valid.

   SYNWORD => SYNWORD                 -- 7-bit output: Word output in the frame where an ECC error has been
                                      -- detected

);

-- End of FRAME_ECC_VIRTEX6_inst instantiation
```

Verilog Instantiation Template

```
// FRAME_ECC_VIRTEX6: Configuration Frame Error Correction
//                    Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

FRAME_ECC_VIRTEX6 #(
   .FARSRC("EFAR"),                // Determines if the output of FAR[23:0] configuration register points to
                                   // the FAR or EFAR. Sets configuration option register bit CTL0[7].
   .FRAME_RBT_IN_FILENAME("NONE")  // This file is output by the ICAP_VIRTEX6 model and it contains Frame
                                   // Data information for the Raw Bitstream (RBT) file. The FRAME_ECC model
                                   // will parse this file, calculate ECC and output any error conditions.
)
FRAME_ECC_VIRTEX6_inst (
   .CRCERROR(CRCERROR),             // 1-bit output: Output indicating a CRC error
```

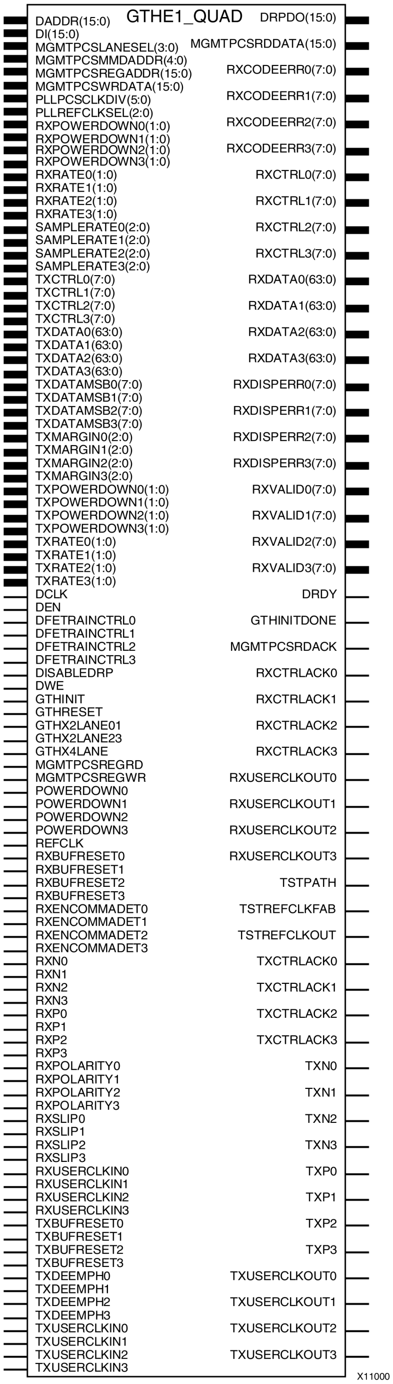
```
   .ECCERROR(ECCERROR),             // 1-bit output: Output indicating an ECC error
   .ECCERRORSINGLE(ECCERRORSINGLE), // 1-bit output: Output Indicating single-bit Frame ECC error detected.
   .FAR(FAR),                       // 24-bit output: Frame Address Register Value output
   .SYNBIT(SYNBIT),                 // 5-bit output: Output bit address of error
   .SYNDROME(SYNDROME),             // 13-bit output: Output location of erroneous bit
   .SYNDROMEVALID(SYNDROMEVALID),   // 1-bit output: Frame ECC output indicating the SYNDROME output is
                                    // valid.

   .SYNWORD(SYNWORD)                // 7-bit output: Word output in the frame where an ECC error has been
                                    // detected

);

// End of FRAME_ECC_VIRTEX6_inst instantiation
```

For More Information

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

GTHE1_QUAD

Primitive: Gigabit Transceiver



Introduction

This design element represents the Virtex®-6 FPGA GTH transceiver. GTH is the highest performance, 10G-optimized configurable transceiver in the Virtex-6 FPGA as part of the HXT family. Refer to *Virtex-6 FPGA GTH Transceivers User Guide* for detailed information regarding this component. The Virtex-6 FPGA GTH Transceivers Wizard is the preferred tool to generate a wrapper to instantiate a GTHE1_QUAD primitive. The Wizard can be found in the Xilinx® CORE Generator™ tool.

Design Entry Method

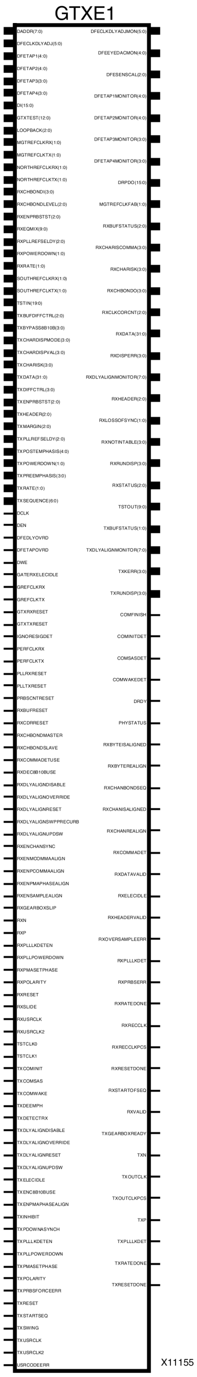
To instantiate this component, use the Virtex-6 FPGA GTH Transceivers Wizard or an associated core containing the component. Xilinx does not recommend direct instantiation of this component.

For More Information

- See the *Virtex-6 FPGA GTH Transceivers User Guide (UG371)*.
- See the *Virtex-6 FPGA User Documentation (User Guides and Data Sheets)*.

GTXE1

Primitive: Gigabit Transceiver



Introduction

This design element represents the Virtex®-6 FPGA RocketIO™ GTX transceiver, a power-efficient and highly configurable transceiver. Refer to *Virtex-6 FPGA RocketIO GTX Transceiver User Guide* for detailed information regarding this component. The Virtex-6 FPGA RocketIO GTX Transceiver Wizard is the preferred tool to generate a wrapper to instantiate a GTXE1 primitive. The Wizard can be found in the Xilinx® CORE Generator™ tool.

Design Entry Method

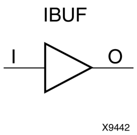
To instantiate this component, use the Virtex-6 FPGA RocketIO GTX Transceiver Wizard or an associated core containing the component. Xilinx does not recommend direct instantiation of this component.

For More Information

- See the *Virtex-6 FPGA RocketIO GTX Transceivers User Guide (UG366)*.
- See the *Virtex-6 FPGA User Documentation (User Guides and Data Sheets)*.

IBUF

Primitive: Input Buffer



Introduction

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

Port Descriptions

Port	Direction	Width	Function
O	Output	1	Buffer 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
--       Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

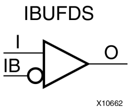
```
// IBUF: Single-ended Input Buffer
//       Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

IBUF #(
   .IBUF_LOW_PWR("TRUE"),  // Low power (TRUE) vs. performrmance (FALSE) setting forr referenced I/O standards
   .IOSTANDARD("DEFAULT")  // Specify the input I/O standard
)IBUF_inst (
   .O(O),     // Buffer output
   .I(I)      // Buffer input (connect directly to top-level port)
);

// End of IBUF_inst instantiation
```

For More Information

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

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
--         Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

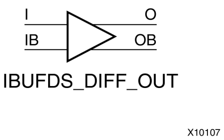
```
// IBUFDS: Differential Input Buffer
//         Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

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

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
DIFF_TERM	Boolean	TRUE, FALSE	FALSE	Specifies the use of the internal differential termination resistance.
IOSTANDARD	String	See Data Sheet.	"DEFAULT"	Assigns an I/O standard to the element.
IBUF_LOW_POWER	Boolean	TRUE, FALSE	FALSE	Allows a trade off of lower power consumption vs. highest performance.

VHDL 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
--                  Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

IBUFDS_DIFF_OUT_inst : IBUFDS_DIFF_OUT
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") -- 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

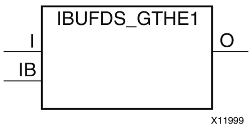
```
// IBUFDS_DIFF_OUT: Differential Input Buffer with Differential Output
//                  Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

IBUFDS_DIFF_OUT #(
   .DIFF_TERM("FALSE"),   // Differential Termination, "TRUE"/"FALSE"
   .IBUF_LOW_PWR("TRUE"), // Low power="TRUE", Highest performrmance="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 *Virtex-6 FPGA User Documentation (User Guides and Data Sheets)*.

IBUFDS_GTHE1

Primitive: Differential Clock Input for the GTH Transceiver Reference Clocks



Introduction

This component is the dedicated differential clock input for the GTH transceiver reference clocks. There is one IBUFGDS_GTHE1 component per GTH quad and it connects directly to the REFCLK pin of the GTHE1_QUAD primitive.

Design Entry Method

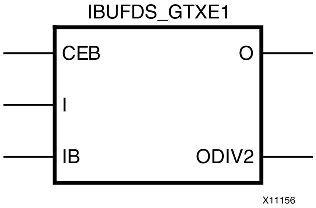
To instantiate this component, use the Virtex-6 FPGA GTH Transceivers Wizard or an associated core containing the component. Xilinx does not recommend direct instantiation of this component.

For More Information

- See the *Virtex-6 FPGA GTH Transceivers User Guide (UG371)*.
- See the *Virtex-6 FPGA User Documentation (User Guides and Data Sheets)*.

IBUFDS_GTXE1

Primitive: Differential Clock Input for the Transceiver Reference Clocks



Introduction

This component is the differential clock input for the transceiver reference clocks. It can also drive other clock resources such as BUFG/MMCM as well as the reference clock inputs of the GT. It typically connects to the MGTREFCLKRX/TX pins of the 4 GTXE1 in the quad associated with the IBUFDS_GTXE1, to the NORTHREFCLKRX/TX of the 4 GTXE1 in the quad above, or to the SOUTHREFCLKRX/TX pins of the 4 GTXE1 in the quad below.

There are multiple destination pins in Virtex®-6 devices that the IBUFDS_GTXE1 element could connect to. If one reference clock on the GT is connected, SW has full control and can route and connect to the GT on any of the pins based on the most optimal route. If multiple clocks are connected to the GT then SW will route each IBUFDS to the indicated pin on the GT. So the O pin on the IBUFDS_GTXE1 connects to either the MGTREFCLKRX/TX or the NORTH/SOUTHREFCLKRX/TX pins on the GT.

Note The RX and TX MUXes can be chosen independently, but the routes are shared on physical silicon.

Design Entry Method

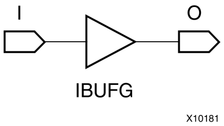
To instantiate this component, use the RocketIO™ wizard or an associated core containing the component. Xilinx does not recommend direct instantiation of this component.

For More Information

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

IBUFG

Primitive: Dedicated Input Clock Buffer



Introduction

The IBUFG is a dedicated input to the device which should be used to connect incoming clocks to the FPGA's global clock routing resources. The IBUFG provides dedicated connections from a top level port to the MMCM or BUFG providing the minimum amount of clock delay and jitter to the device. The IBUFG input can only be driven by the clock capable (CC) or 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
--        Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

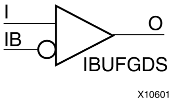
```
// IBUFG: Single-ended global clock input buffer
//        Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

IBUFG #(
   .IBUF_LOW_PWR("TRUE"),  // Low power (TRUE) vs. performrmance (FALSE) setting forr referenced I/O standards
   .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 *Virtex-6 FPGA User Documentation (User Guides and Data Sheets)*.

IBUFGDS

Primitive: Differential Signaling Dedicated Input Clock Buffer and Optional Delay



Introduction

This design element is a dedicated differential signaling input buffer for connection to the clock buffer (BUFG) or 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
--          Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

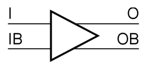
```
// IBUFGDS: Differential Global Clock Input Buffer
//          Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

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

IBUFGDS_DIFF_OUT

Primitive: Differential Signaling Input Buffer with Differential Output



IBUFGDS_DIFF_OUT

X12011

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.
IBUF_LOW_PWR	Boolean	TRUE, FALSE	FALSE	Allows a trade off of lower power consumption vs. highest performance

VHDL 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
--                   Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

IBUFGDS_DIFF_OUT_inst : IBUFGDS_DIFF_OUT
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") -- 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

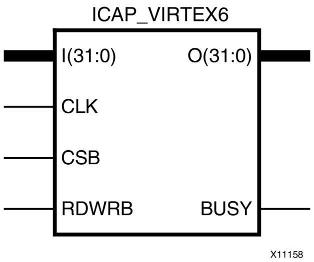
```
// IBUFGDS_DIFF_OUT: Differential Global Clock Buffer with Differential Output
//                   Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

IBUFGDS_DIFF_OUT #(
   .DIFF_TERM("FALSE"),   // Differential Termination, "TRUE"/"FALSE"
   .IBUF_LOW_PWR("TRUE"), // Low power="TRUE", Highest performrmance="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 *Virtex-6 FPGA User Documentation (User Guides and Data Sheets)*.

ICAP_VIRTEX6

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.
CLK	Input	1	Clock Input.
CSB	Input	1	Active-Low ICAP Enable.
I[31:0]	Input	32	Configuration data input bus.
O[31:0]	Output	32	Configuration data output bus.
RDWRB	Input	1	Read/Write Select.

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'h04244093, 32'h042CA093, 32'h042CC093, 32'h042C4093, 32'h042D0093, 32'h0423A093, 32'h0424A093, 32'h0424C093, 32'h04240093, 32'h04248093, 32'h04250093, 32'h04252093, 32'h04256093, 32'h04286093, 32'h04288093	32'h04244093	Specifies the pre-programmed Device ID value to be used for simulation purposes.
ICAP_WIDTH	String	"X8", "X16", "X32"	"X8"	Specifies the input and output data width to be used with the ICAP_VIRTEX6.
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_VIRTEX6: Internal Configuration Access Port
--               Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

ICAP_VIRTEX6_inst : ICAP_VIRTEX6
generic map (
   DEVICE_ID => X"4244093",     -- Specifies the pre-programmed Device ID value
   ICAP_WIDTH => "X8",          -- Specifies the input and output data width to be used with the
                                -- ICAP_VIRTEX6.
   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,         -- 32-bit output: Configuration data output bus
   CLK => CLK,     -- 1-bit input: Clock Input
   CSB => CSB,     -- 1-bit input: Active-Low ICAP input Enable
   I => I,         -- 32-bit input: Configuration data input bus
   RDWRB => RDWRB  -- 1-bit input: Read/Write Select input
);

-- End of ICAP_VIRTEX6_inst instantiation
```

Verilog Instantiation Template

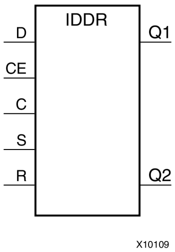
```
// ICAP_VIRTEX6: Internal Configuration Access Port
//               Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

ICAP_VIRTEX6 #(
   .DEVICE_ID(0'h4244093),     // Specifies the pre-programmed Device ID value
   .ICAP_WIDTH("X8"),          // Specifies the input and output data width to be used with the
                               // ICAP_VIRTEX6.
   .SIM_CFG_FILE_NAME("NONE")  // Specifies the Raw Bitstream (RBT) file to be parsed by the simulation
                               // model
)
ICAP_VIRTEX6_inst (
   .BUSY(BUSY),   // 1-bit output: Busy/Ready output
   .O(O),         // 32-bit output: Configuration data output bus
   .CLK(CLK),     // 1-bit input: Clock Input
   .CSB(CSB),     // 1-bit input: Active-Low ICAP input Enable
   .I(I),         // 32-bit input: Configuration data input bus
   .RDWRB(RDWRB)  // 1-bit input: Read/Write Select input
);

// End of ICAP_VIRTEX6_inst instantiation
```

For More Information

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

IDDR

Primitive: Input Dual Data-Rate Register



Introduction

This design element is a dedicated input register designed to receive external dual data rate (DDR) signals into Xilinx® FPGAs.The IDDR is available with modes that present the data to the FPGA fabric at the time and clock edge they are captured, or on the same clock edge. This feature allows you to avoid additional timing complexities and resource usage.

- **OPPOSITE_EDGE mode -** Data is recovered in the classic DDR methodology. Given a DDR data and clock at pin D and C respectively, Q1 changes after every positive edge of clock C, and Q2 changes after every negative edge of clock C.
- **SAME_EDGE mode -** Data is still recovered by opposite edges of clock C. However, an extra register has been placed in front of the negative edge data register. This extra register is clocked with positive clock edge of clock signal C. As a result, DDR data is now presented into the FPGA fabric at the same clock edge. However, because of this feature, the data pair appears to be "separated." Q1 and Q2 no longer have pair 1 and 2. Instead, the first pair presented is Pair 1 and DONT_CARE, followed by Pair 2 and 3 at the next clock cycle.
- **SAME_EDGE_PIPELINED mode -** Recovers data in a similar fashion as the SAME_EDGE mode. In order to avoid the "separated" effect of the SAME_EDGE mode, an extra register has been placed in front of the positive edge data register. A data pair now appears at the Q1 and Q2 pin at the same time. However, using this mode costs you an additional cycle of latency for Q1 and Q2 signals to change.

IDDR also works with the SelectIO™ features, such as the IODELAY.

Note For high speed interfaces, the IDDR_2CLK component can be used to specify two independent clocks to capture the data. Use this component when the performance requirements of the IDDR are not adequate, since the IDDR_2CLK requires more clocking resources and can imply placement restrictions that are not necessary when using the IDDR component.

Port Descriptions

Port	Direction	Width	Function
Q1 - Q2	Output	1	These pins are the IDDR output that connects to the FPGA fabric. Q1 is the first data pair and Q2 is the second data pair.
C	Input	1	Clock input pin.
CE	Input	1	When asserted Low, this port disables the output clock at port O.
D	Input	1	This pin is where the DDR data is presented into the IDDR module. This pin connects to a top-level input or bi-directional port, and IODELAY configured for an input delay or to an appropriate input or bidirectional buffer.
R	Input	1	Active high reset forcing Q1 and Q2 to a logic zero. Can be synchronous or asynchronous based on the SRTYPE attribute.
S	Input	1	Active high reset forcing Q1 and Q2 to a logic one. Can be synchronous or asynchronous based on the SRTYPE attribute.

Design Entry Method

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

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
DDR_CLK_EDGE	String	"OPPOSITE_EDGE", "SAME_EDGE", "SAME_EDGE_ PIPELINED"	"OPPOSITE_ EDGE"	Sets the IDDR mode of operation with respect to clock edge.
INIT_Q1	Binary	0, 1	0	Initial value on the Q1 pin after configuration startup or when GSR is asserted.
INIT_Q2	Binary	0, 1	0	Initial value on the Q2 pin after configuration startup or when GSR is asserted.
SRTYPE	String	"SYNC" or "ASYNC"	"SYNC"	Set/reset type selection. "SYNC" specifies the behavior of the reset (R) and set (S) pins to be synchronous to the positive edge of the C clock pin. "ASYNC" specifies an asynchronous set/reset 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;


-- IDDR: Double Data Rate Input Register with Set, Reset
--       and Clock Enable.
--       Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

IDDR_inst : IDDR
generic map (
   DDR_CLK_EDGE => "OPPOSITE_EDGE", -- "OPPOSITE_EDGE", "SAME_EDGE"
                                    -- or "SAME_EDGE_PIPELINED"
   INIT_Q1 => '0', -- Initial value of Q1: '0' or '1'
   INIT_Q2 => '0', -- Initial value of Q2: '0' or '1'
   SRTYPE => "SYNC") -- Set/Reset type: "SYNC" or "ASYNC"
port map (
   Q1 => Q1, -- 1-bit output for positive edge of clock
   Q2 => Q2, -- 1-bit output for negative edge of clock
   C => C,   -- 1-bit clock input
   CE => CE, -- 1-bit clock enable input
   D => D,   -- 1-bit DDR data input
   R => R,   -- 1-bit reset
   S => S    -- 1-bit set
   );

-- End of IDDR_inst instantiation
```

Verilog Instantiation Template

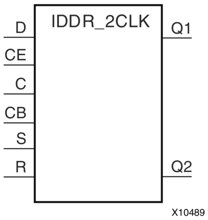
```
// IDDR: Input Double Data Rate Input Register with Set, Reset
//       and Clock Enable.
//       Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

IDDR #(
   .DDR_CLK_EDGE("OPPOSITE_EDGE"), // "OPPOSITE_EDGE", "SAME_EDGE"
                                   //    or "SAME_EDGE_PIPELINED"
   .INIT_Q1(1'b0), // Initial value of Q1: 1'b0 or 1'b1
   .INIT_Q2(1'b0), // Initial value of Q2: 1'b0 or 1'b1
   .SRTYPE("SYNC") // Set/Reset type: "SYNC" or "ASYNC"
) IDDR_inst (
   .Q1(Q1), // 1-bit output forr positive edge of clock
   .Q2(Q2), // 1-bit output forr negative edge of clock
   .C(C),   // 1-bit clock input
   .CE(CE), // 1-bit clock enable input
   .D(D),   // 1-bit DDR data input
   .R(R),   // 1-bit reset
   .S(S)    // 1-bit set
);

// End of IDDR_inst instantiation
```

For More Information

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

IDDR_2CLK

Primitive: Input Dual Data-Rate Register with Dual Clock Inputs



Introduction

This design element is a dedicated input register designed to receive external dual data rate (DDR) signals into Xilinx® FPGAs. In general, you should only use the IDDR_2CLK for applications in which two clocks are required to capture the rising and falling data for DDR applications.

- **OPPOSITE_EDGE mode -** Data is presented in the classic DDR methodology. Given a DDR data and clock at pin D and C respectively, Q1 changes after every positive edge of clock C, and Q2 changes after every positive edge of clock CB.
- **SAME_EDGE mode -** Data is still presented by positive edges of each clock. However, an extra register has been placed in front of the CB clocked data register. This extra register is clocked with positive clock edge of clock signal C. As a result, DDR data is now presented into the FPGA fabric at the positive edge of clock C. However, because of this feature, the data pair appears to be "separated." Q1 and Q2 no longer have pair 1 and 2. Instead, the first pair presented is Pair 1 and DON'T CARE, followed by Pair 2 and 3 at the next clock cycle.
- **SAME_EDGE_PIPELINED mode -** Presents data in a similar fashion as the SAME_EDGE mode. In order to avoid the "separated" effect of the SAME_EDGE mode, an extra register has been placed in front of the C clocked data register. A data pair now appears at the Q1 and Q2 pin at the same time during the positive edge of C. However, using this mode, costs you an additional cycle of latency for Q1 and Q2 signals to change.

IDDR also works with SelectIO™ features, such as the IODELAY.

Port Descriptions

Port	Direction	Width	Function
Q1 : Q2	Output	1	These pins are the IDDR output that connects to the FPGA fabric. Q1 is the first data pair and Q2 is the second data pair.
C	Input	1	Primary clock input pin used to capture the positive edge data.
CB	Input	1	Secondary clock input pin (typically 180 degrees out of phase with the primary clock) used to capture the negative edge data.
CE	Input	1	When asserted Low, this port disables the output clock at port O.
D	Input	1	This pin is where the DDR data is presented into the IDDR module. This pin connects to a top-level input or bi-directional port, and IODELAY configured for an input delay or to an appropriate input or bidirectional buffer.
R	Input	1	Active high reset forcing Q1 and Q2 to a logic zero. Can be synchronous or asynchronous based on the SRTYPE attribute.

Port	Direction	Width	Function
S	Input	1	Active high reset forcing Q1 and Q2 to a logic one. Can be synchronous or asynchronous based on the SRTYPE attribute.

Design Entry Method

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

- Connect the C pin to the appropriate clock source, representing the positive clock edge and CB to the clock source representing the negative clock edge.
- Connect the D pin to the top-level input, or bidirectional port, an IODELAY, or an instantiated input or bidirectional buffer.
- The Q1 and Q2 pins should be connected to the appropriate data sources.
- CE should be tied high when not used, or connected to the appropriate clock enable logic.
- R and S pins should be tied low, if not used, or to the appropriate set or reset generation logic.
- Set all attributes to the component to represent the desired behavior.
- Always instantiate this component in pairs with the same clocking, and to LOC those to the appropriate P and N I/O pair in order not to sacrifice possible I/O resources.
- Always instantiate this component in the top-level hierarchy of your design, along with any other instantiated I/O components for the design. This helps facilitate hierarchical design flows/practices.
- To minimize CLK skew, both CLK and CLKB should come from global routing (DCM / MMCM) and not from the local inversion. DCM / MMCM de-skews these clocks whereas the local inversion adds skew.

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
DDR_CLK_EDGE	String	"OPPOSITE_EDGE", "SAME_EDGE" "SAME_EDGE_ PIPELINED"	"OPPOSITE_ EDGE"	DDR clock mode recovery mode selection. See Introduction for more explanation.
INIT_Q1	Binary	0, 1	0	Initial value on the Q1 pin after configuration startup or when GSR is asserted.
INIT_Q2	Binary	0, 1	0	Initial value on the Q2 pin after configuration startup or when GSR is asserted.
SRTYPE	String	"SYNC" or "ASYNC"	"SYNC"	Set/reset type selection. "SYNC" specifies the behavior of the reset (R) and set (S) pins to be synchronous to the positive edge of the C clock pin. "ASYNC" specifies an asynchronous set/reset function.

VHDL Instantiation Template

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

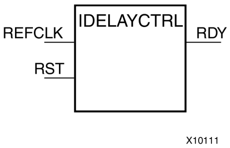
```
Library UNISIM;
use UNISIM.vcomponents.all;


-- IDDR_2CLK: Dual-Clock, Input Double Data Rate Input Register with
--            Set, Reset and Clock Enable.
--            Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

IDDR_2CLK_inst : IDDR_2CLK
generic map (
   DDR_CLK_EDGE => "OPPOSITE_EDGE", -- "OPPOSITE_EDGE", "SAME_EDGE"
                                    -- or "SAME_EDGE_PIPELINED"
   INIT_Q1 => '0', -- Initial value of Q1: '0' or '1'
   INIT_Q2 => '0', -- Initial value of Q2: '0' or '1'
   SRTYPE => "SYNC") -- Set/Reset type: "SYNC" or "ASYNC"
port map (
   Q1 => Q1, -- 1-bit output for positive edge of clock
   Q2 => Q2, -- 1-bit output for negative edge of clock
   C => C,   -- 1-bit primary clock input
   CB => CB, -- 1-bit secondary clock input
   CE => CE, -- 1-bit clock enable input
   D => D,   -- 1-bit DDR data input
   R => R,   -- 1-bit reset
   S => S    -- 1-bit set
   );

-- End of IDDR_2CLK_inst instantiation
```

Verilog Instantiation Template

```
// IDDR_2CLK: Dual-Clock, Input Double Data Rate Input Register with
//            Set, Reset and Clock Enable.
//            Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

IDDR_2CLK #(
   .DDR_CLK_EDGE("OPPOSITE_EDGE"), // "OPPOSITE_EDGE", "SAME_EDGE"
                                   //    or "SAME_EDGE_PIPELINED"
   .INIT_Q1(1'b0), // Initial value of Q1: 1'b0 or 1'b1
   .INIT_Q2(1'b0), // Initial value of Q2: 1'b0 or 1'b1
   .SRTYPE("SYNC") // Set/Reset type: "SYNC" or "ASYNC"
) IDDR_2CLK_inst (
   .Q1(Q1), // 1-bit output forr positive edge of clock
   .Q2(Q2), // 1-bit output forr negative edge of clock
   .C(C),   // 1-bit primary clock input
   .CB(CB), // 1-bit secondary clock input
   .CE(CE), // 1-bit clock enable input
   .D(D),   // 1-bit DDR data input
   .R(R),   // 1-bit reset
   .S(S)    // 1-bit set
);

// End of IDDR_2CLK_inst instantiation
```

IDELAYCTRL

Primitive: IDELAY Tap Delay Value Control



Introduction

This design element must be instantiated when using the IODELAYE1. This occurs when the IDELAY or ISERDES primitive is instantiated with the IOBDELAY_TYPE attribute set to Fixed or Variable. The IDELAYCTRL module provides a voltage bias, independent of process, voltage, and temperature variations to the tap-delay line using a fixed-frequency reference clock, REFCLK. This enables very accurate delay tuning.

Port Descriptions

Port	Direction	Width	Function
RDY	Output	1	Indicates the validity of the reference clock input, REFCLK. When REFCLK disappears (i.e., REFCLK is held High or Low for one clock period or more), the RDY signal is deasserted.
REFCLK	Input	1	Provides a voltage bias, independent of process, voltage, and temperature variations, to the tap-delay lines in the IOBs. The frequency of REFCLK must be 200 MHz to guarantee the tap-delay value specified in the applicable data sheet.
RST	Input	1	Resets the IDELAYCTRL circuitry. The RST signal is an active-high asynchronous reset. To reset the IDELAYCTRL, assert it High for at least 50 ns.

RST (Module reset) - Resets the IDELAYCTRL circuitry. The RST signal is an active-high asynchronous reset. To reset the IDELAYCTRL, assert it High for at least 50 ns.

REFCLK (Reference Clock) - Provides a voltage bias, independent of process, voltage, and temperature variations, to the tap-delay lines in the IOBs. The frequency of REFCLK must be 200 MHz to guarantee the tap-delay value specified in the applicable data sheet.

RDY (Ready Output) - Indicates the validity of the reference clock input, REFCLK. When REFCLK disappears (i.e., REFCLK is held High or Low for one clock period or more), the RDY signal is deasserted.

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;


-- IDELAYCTRL : Input Delay Element Control
--         Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

IDELAYCTRL_inst : IDELAYCTRL
port map (
   RDY => RDY,       -- 1-bit output indicates validity of the REFCLK
   REFCLK => REFCLK, -- 1-bit reference clock input
   RST => RST        -- 1-bit reset input
);

-- End of IDELAYCTRL_inst instantiation
```

Verilog Instantiation Template

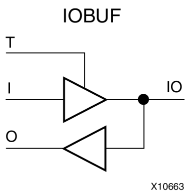
```
// IDELAYCTRL: IDELAY Tap Delay Value Control
//             Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

(* IODELAY_GROUP = "<iodelay_group_name>" *) // Specifies group name forr associated IODELAYs and IDELAYCTRL
IDELAYCTRL IDELAYCTRL_inst (
   .RDY(RDY),       // 1-bit Ready output
   .REFCLK(REFCLK), // 1-bit Reference clock input
   .RST(RST)        // 1-bit Reset input
);

// End of IDELAYCTRL_inst instantiation
```

For More Information

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

IOBUF

Primitive: Bi-Directional Buffer



Introduction

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

Logic Table

Inputs		Bidirectional	Outputs
T	**I**	**IO**	**O**
1	X	Z	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
--        Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

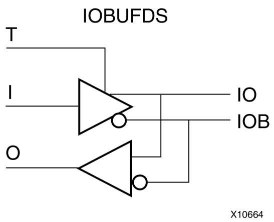
```
// IOBUF: Single-ended Bi-directional Buffer
//        All devices
// Xilinx HDL Libraries Guide, version 14.2

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

IOBUFDS

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



Introduction

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

Logic Table

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

Port Descriptions

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
--          Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

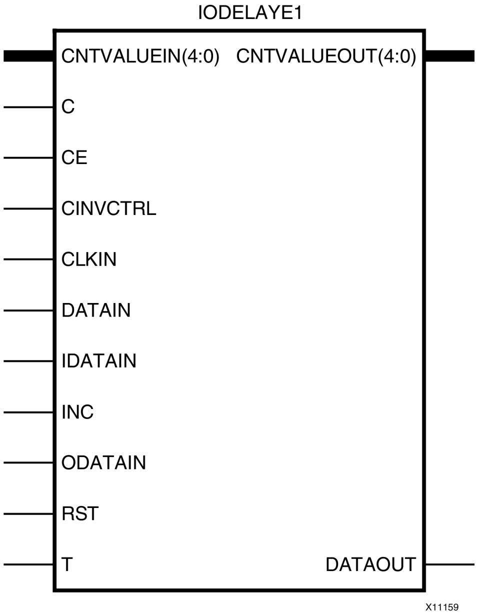
```
// IOBUFDS: Differential Bi-directional Buffer
//          Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

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

IODELAYE1

Primitive: Input and Output Fixed or Variable Delay Element



Introduction

Every I/O block contains a programmable absolute delay element called IODELAYE1. The IODELAYE1 can be connected to an input register/ISERDESE1 or output register/OSERDESE1 block or both. IODELAYE1 is a 31-tap, wraparound, delay element with a calibrated tap resolution. Refer to the Virtex-6 FPGA Data Sheet for delay values. It can be applied to the combinatorial input path, registered input path, combinatorial output path, or registered output path. It can also be accessed directly in the FPGA logic. IODELAYE1 allows incoming signals to be delayed on an individual basis. The tap delay resolution is varied by selecting an IDELAYCTRL reference clock from the range specified in the Virtex-6 FPGA Data Sheet. The IODELAYE1 resource can function as an input, output, or bidirectional delay.

Port Descriptions

Port	Direction	Width	Function
C	Input	1	Clock input used in VARIABLE or VAR_LOADABLE mode.
CE	Input	1	Active high enable increment/decrement function.
CINVCTRL	Input	1	Dynamically inverts the Clock (C) polarity.
CLKIN	Input	1	Clock Access into the IODELAY (from the IO CLKMUX).
CNTVALUEIN[4:0]	Input	5	Tap counter value from FPGA logic for dynamically loadable tap value.
CNTVALUEOUT[4:0]	Output	5	Tap counter value going to FPGA logic for monitoring tap value
DATAIN	Input	1	The DATAIN input is directly driven by the FPGA logic providing a logic accessible delay line. The data is driven back into the FPGA logic through the DATAOUT port with a delay set by the IDELAY_VALUE. DATAIN can be locally inverted. The data cannot be driven to an IOB.
DATAOUT	Output	1	Delayed data from the three data input ports. DATAOUT connects to the FPGA logic (IDELAY mode), or an IOB (ODELAY mode) or both (bidirectional delay mode). If used in the bidirectional delay

Port	Direction	Width	Function
			mode, the T port dynamically switches between the IDATAIN and ODATAIN paths providing an alternating input/output delay based on the direction indicated by the 3-state signal T from the OLOGIC block.
IDATAIN	Input	1	The IDATAIN input is driven by its associated IOB. In IDELAY mode the data can be driven to either an ILOGIC/ISERDES block, directly into the FPGA logic, or to both through the DATAOUT port with a delay set by the IDELAY_VALUE.
INC	Input	1	Increment/decrement number of tap delays.
ODATAIN	Input	1	The ODATAIN input is driven by OLOGIC/OSERDES. In ODELAY mode, the ODATAIN drives the DATAOUT port which is connected to an IOB with a delay set by the ODELAY_VALUE.
RST	Input	1	When in VARIABLE mode, the IODELAYE1 reset signal, RST, resets the delay element to a value set by the IDELAY_VALUE or ODELAY_VALUE attribute. If these attributes are not specified, a value of zero is assumed. The RST signal is an active-High reset and is synchronous to the input clock signal (C). When in VAR_LOADABLE mode, the IODELAYE1 reset signal, RST, resets the delay element to a value set by the CNTVALUEIN. The value present at CNTVALUEIN[4:0] will be the new tap value. As a results of this functionality the IDELAY_VALUE and ODELAY_VALUE attribute is ignored.
T	Input	1	This is the 3-state input control port. For bidirectional operation, the T pin signal also controls the T pin of the OBUFT. Tie high for input-only or internal delay or tie low for output only.

Design Entry Method

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

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
CINVCTRL_SEL	Boolean	FALSE, TRUE	FALSE	Dynamically inverts the Clock (C) polarity.
DELAY_SRC	String	"CLKIN", "DATAIN", "I", "IO", "O"	"I"	Specifies the source to the IODELAY component. • CLKIN - IODELAYE1 input is CLKIN. • DATAIN - Not connected to any port (internal mode). • I - Connects directly to an input port or IBUF (input mode). • IO - Connects to a port. • O - Connects to an output port or OBUF (output mode).
HIGH_PERFORMANCE_MODE	Boolean	TRUE, FALSE	TRUE	When TRUE, this attribute reduces the output jitter. When FALSE, reduces power consumption. The difference in power

Attribute	Data Type	Allowed Values	Default	Description
				consumption is quantified in the XPower Estimator (XPE) tool.
IDELAY_TYPE	String"	"DEFAULT", "FIXED", "VARIABLE", VAR_LOADABLE"	"DEFAULT"	Sets the type of tap delay line. DEFAULT delay guarantees zero hold times. FIXED delay sets a static delay value. VAR_LOADABLE dynamically loads tap values. VARIABLE delay dynamically adjusts the delay value.
IDELAY_VALUE	Integer	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31	0	Specifies the fixed number of delay taps in fixed mode or the initial starting number of taps in VARIABLE mode (input path). When IDELAY_TYPE is set to VAR_LOADABLE mode, this value is ignored.
ODELAY_TYPE	String	"FIXED, "VARIABLE", "VAR_LOADABLE"	"FIXED"	Specifies a fixed, variable or default (eliminate hold time) output delay.
ODELAY_VALUE	Integer	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31	0	Specifies the fixed number of delay taps in fixed mode or the initial starting number of taps in VARIABLE mode (output path). When IDELAY_TYPE is set to VAR_LOADABLE mode, this value is ignored.
REFCLK_ FREQUENCY	1 significant digit FLOAT	190.0 to 210.1 and 290.0 to 310.0	200.0	Sets the tap value (in MHz) used by the timing analyzer for static timing analysis and functional/timing simulation. The frequency of REFCLK must be within the given datasheet range to guarantee the tap-delay value and performance.
SIGNAL_PATTERN	String	"DATA", "CLOCK"	"DATA"	Causes the timing analyzer to account for the appropriate amount of delay-chain jitter in the data or clock path.

VHDL Instantiation Template

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

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


-- IODELAYE1: Input and Output Fixed or Variable Delay Element
--            Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

IODELAYE1_inst : IODELAYE1
generic map (
   CINVCTRL_SEL => FALSE,          -- Enable dynamic clock inversion ("TRUE"/"FALSE")
   DELAY_SRC => "I",                -- Delay input ("I", "CLKIN", "DATAIN", "IO", "O")
   HIGH_PERFORMANCE_MODE => TRUE, -- Reduced jitter ("TRUE"), Reduced power ("FALSE")
   IDELAY_TYPE => "DEFAULT",        -- "DEFAULT", "FIXED", "VARIABLE", or "VAR_LOADABLE"
   IDELAY_VALUE => 0,               -- Input delay tap setting (0-32)
   ODELAY_TYPE => "FIXED",          -- "FIXED", "VARIABLE", or "VAR_LOADABLE"
   ODELAY_VALUE => 0,               -- Output delay tap setting (0-32)
   REFCLK_FREQUENCY => 200.0,       -- IDELAYCTRL clock input frequency in MHz
   SIGNAL_PATTERN => "DATA"         -- "DATA" or "CLOCK" input signal
)
port map (
   CNTVALUEOUT => CNTVALUEOUT, -- 5-bit output - Counter value for monitoring purpose
   DATAOUT => DATAOUT,         -- 1-bit output - Delayed data output
   C => C,                     -- 1-bit input - Clock input
   CE => CE,                   -- 1-bit input - Active high enable increment/decrement function
   CINVCTRL => CINVCTRL,       -- 1-bit input - Dynamically inverts the Clock (C) polarity
```

```
   CLKIN => CLKIN,              -- 1-bit input - Clock Access into the IODELAY
   CNTVALUEIN => CNTVALUEIN,    -- 5-bit input - Counter value for loadable counter application
   DATAIN => DATAIN,            -- 1-bit input - Internal delay data
   IDATAIN => IDATAIN,          -- 1-bit input - Delay data input
   INC => INC,                  -- 1-bit input - Increment / Decrement tap delay
   ODATAIN => ODATAIN,          -- 1-bit input - Data input for the output datapath from the device
   RST => RST,                  -- 1-bit input - Active high, synchronous reset, resets delay chain to IDELAY_VALUE/
                                -- ODELAY_VALUE tap. If no value is specified, the default is 0.
   T => T                       -- 1-bit input - 3-state input control. Tie high for input-only or internal delay or
                                -- tie low for output only.

);

-- End of IODELAYE1_inst instantiation
```

Verilog Instantiation Template

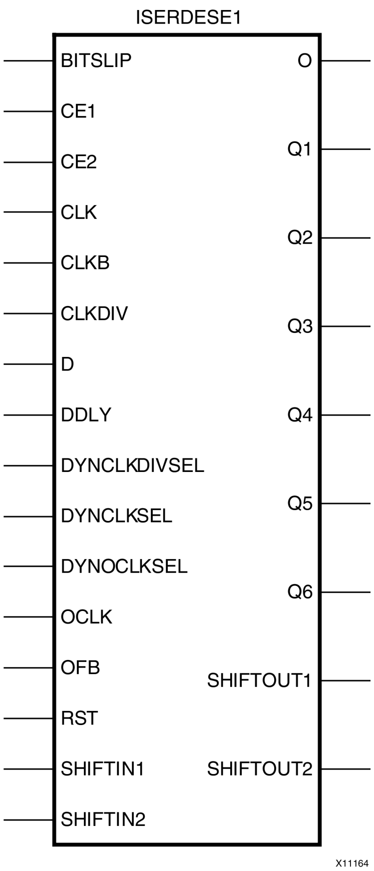
```
// IODELAYE1: Input / Output Fixed or Variable Delay Element
//            Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

(* IODELAY_GROUP = "<iodelay_group_name>" *) // Specifies group name forr associated IODELAYs and IDELAYCTRL
IODELAYE1 #(
   .CINVCTRL_SEL("FALSE"),          // Enable dynamic clock inversion ("TRUE"/"FALSE")
   .DELAY_SRC("I"),                 // Delay input ("I", "CLKIN", "DATAIN", "IO", "O")
   .HIGH_PERforRMANCE_MODE("FALSE"), // Reduced jitter ("TRUE"), Reduced power ("FALSE")
   .IDELAY_TYPE("DEFAULT"),         // "DEFAULT", "FIXED", "VARIABLE", or "VAR_LOADABLE"
   .IDELAY_VALUE(0),                // Input delay tap setting (0-32)
   .ODELAY_TYPE("FIXED"),           // "FIXED", "VARIABLE", or "VAR_LOADABLE"
   .ODELAY_VALUE(0),                // Output delay tap setting (0-32)
   .REFCLK_FREQUENCY(200.0),        // IDELAYCTRL clock input frequency in MHz
   .SIGNAL_PATTERN("DATA")          // "DATA" or "CLOCK" input signal
)
IODELAYE1_inst (
   .CNTVALUEOUT(CNTVALUEOUT), // 5-bit output - Counter value forr monitoring purpose
   .DATAOUT(DATAOUT),         // 1-bit output - Delayed data output
   .C(C),                     // 1-bit input - Clock input
   .CE(CE),                   // 1-bit input - Active high enable increment/decrement function
   .CINVCTRL(CINVCTRL),       // 1-bit input - Dynamically inverts the Clock (C) polarity
   .CLKIN(CLKIN),             // 1-bit input - Clock Access into the IODELAY
   .CNTVALUEIN(CNTVALUEIN),   // 5-bit input - Counter value forr loadable counter application
   .DATAIN(DATAIN),           // 1-bit input - Internal delay data
   .IDATAIN(IDATAIN),         // 1-bit input - Delay data input
   .INC(INC),                 // 1-bit input - Increment / Decrement tap delay
   .ODATAIN(ODATAIN),         // 1-bit input - Data input forr the output datapath from the device
   .RST(RST),                 // 1-bit input - Active high, synchronous reset, resets delay chain to IDELAY_VALUE/
                              // ODELAY_VALUE tap. If no value is specified, the default is 0.
   .T(T)                      // 1-bit input - 3-state input control. Tie high forr input-only or internal delay or
                              // tie low forr output only.

);

// End of IODELAYE1_inst instantiation
```

For More Information

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

ISERDESE1

Primitive: Input SERial/DESerializer



Introduction

This design element is a dedicated serial-to-parallel converter with specific clocking and logic features designed to facilitate the implementation of high-speed source-synchronous applications. It avoids the additional timing complexities encountered when designing deserializers in the FPGA fabric.

Port Descriptions

Port	Direction	Width	Function
BITSLIP	Input	1	The BITSLIP pin performs a Bitslip operation synchronous to CLKDIV when asserted (active High). Subsequently, the data seen on the Q1 to Q6 output ports will shift, as in a barrel-shifter operation, one position every time Bitslip is invoked (DDR operation is different from SDR).
CE1	Input	1	Data register clock enable.
CE2	Input	1	Data register clock enable.
CLK	Input	1	Primary clock input pin used.
CLKB	Input	1	The high-speed secondary clock input (CLKB) is used to clock in the input serial data stream. In any mode other than MEMORY_QDR, connect CLKB to an inverted version of CLK. In

Port	Direction	Width	Function
			MEMORY_QDR mode CLKB should be connected to a unique, phase shifted clock
CLKDIV	Input	1	Divided clock to be used for parallelized data.
D	Input	1	Input data to be connected directly to the top-level input or I/O port of the design or to an IODELAY component if additional input delay control is desired.
DDLY	Input	1	Serial input from IODELAY.
DYNCLKDIVSEL	Input	1	Dynamically select CLKDIV inversion.
DYNCLKSEL	Input	1	Dynamically select CLK and CLKB inversion.
O	Output	1	Combinatorial output.
OCLK	Input	1	High speed output clock typically used for memory interfaces.
OCLKB	Input	1	Used for Async Oversampling.
OFB	Input	1	The output feedback port (OFB) is the serial (high-speed) data output port of the OSERDESE1 or the bypassed version of the CLKPERF. When the attribute ODELAYUSED is set to 0, the OFB port can be used to send out serial data to the ISERDESE1. When the attribute ODELAYUSED is set to 1 and the OSERDESE1 is in MEMORY_DDR3 mode, the OFB port can be used to link the high-performance clock input (CLKPERF) to the IODELAYE1.
Q1 - Q6	Output	1	The output ports Q1 to Q6 are the registered outputs of the ISERDESE1 module. One ISERDESE1 block can support up to six bits (i.e., a 1:6 deserialization). Bit widths greater than six (up to 10) can be supported.
RST	Input	1	Active High asynchronous reset signal for the registers of the SERDES.
SHIFTIN1/ SHIFTIN2	Input	1	If ISERDES_MODE="SLAVE" connect to the master ISERDES_NODELAY IDATASHIFTOUT1/2 outputs. This pin must be grounded.
SHIFTOUT1/ SHIFTOUT2	Output	1	If ISERDES_MODE="MASTER" and two ISERDES_NODELAY are to be cascaded, connect to the slave ISERDES_NODELAY IDATASHIFTIN1/2 inputs.

Design Entry Method

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

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
DATA_RATE	String	"DDR", "SDR"	"DDR"	Enables incoming data stream to be processed as SDR or DDR data.
DATA_WIDTH	Integer	4, 2, 3, 5, 6, 7, 8, 10	4	Defines the width of the serial-to-parallel converter. The legal value depends on the DATA_RATE attribute (SDR or DDR). • If DATA_RATE = DDR, value is limited to 4, 6, 8, or 10. • If DATA_RATE = SDR, value is limited to 2, 3, 4, 5, 6, 7, or 8.
DYN_CLKDIV_INV_EN	Boolean	FALSE, TRUE	FALSE	Enables DYNCLKDIVINVSEL inversion when TRUE and disables HDL inversions on CLKDIV pin.
DYN_CLK_INV_EN	Boolean	FALSE, TRUE	FALSE	Enables DYNCLKINVSEL inversion when TRUE and disables HDL inversions on CLK and CLKB pins.
INIT_Q1 - INIT_Q4	Binary	1'b0 to 1'b1	1'b0	Defines the initial value on the Q outputs.
INTERFACE_TYPE	String	"MEMORY", "MEMORY_DDR3", "MEMORY_QDR", "NETWORKING"	"MEMORY"	Memory or Networking interface type.
IOBDELAY	String	"NONE", "BOTH", "IBUF", "IFD"	"NONE"	Defines input sources for ISERDES module.
NUM_CE	Integer	2, 1	2	Specifies the number of clock enables.
OFB_USED	Boolean	FALSE, TRUE	FALSE	The OFB port in the ISERDESE1 and OSERDESE1 can be used to feed the data transmitted on the OSERDESE1 back to the ISERDESE1. This feature is enabled when the attribute OFB_USED = TRUE. The OSERDESE1 and ISERDESE1 must have the same DATA_RATE and DATA_WIDTH setting for the feedback to give the correct data. When using the ISERDESE1 and OSERDESE1 in width expansion mode only, connect the master OSERDESE1 to the master ISERDESE1. By using the ISERDESE1 as a feedback port, it can not be used as an input for external data. **Note** OFB_USED should be set to FALSE even if the OFB is used but only for the delaying of the OSERDES output

Attribute	Data Type	Allowed Values	Default	Description
SERDES_MODE	String	"MASTER", "SLAVE"	"MASTER"	Specify whether the ISERDES is operating in master or slave modes when cascaded width expansion.
SRVAL_Q1 - SRVAL_Q4	Binary	1'b0 to 1'b1	1'b0	Defines the value of Q outputs when the SR is invoked.

VHDL Instantiation Template

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

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


-- ISERDESE1: Input SERial/DESerializer
--            Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

ISERDESE1_inst : ISERDESE1
generic map (
   DATA_RATE => "DDR",          -- "SDR" or "DDR"
   DATA_WIDTH => 4,             -- Parallel data width (2-8, 10)
   DYN_CLKDIV_INV_EN => FALSE, -- Enable DYNCLKDIVINVSEL inversion (TRUE/FALSE)
   DYN_CLK_INV_EN => FALSE,     -- Enable DYNCLKINVSEL inversion (TRUE/FALSE)
   -- INIT_Q1 - INIT_Q4: Initial value on the Q outputs (0/1)
   INIT_Q1 => '0',
   INIT_Q2 => '0',
   INIT_Q3 => '0',
   INIT_Q4 => '0',
   INTERFACE_TYPE => "MEMORY", -- "MEMORY", "MEMORY_DDR3", "MEMORY_QDR", "NETWORKING", or "OVERSAMPLE"
   IOBDELAY => "NONE",          -- "NONE", "IBUF", "IFD", "BOTH"
   NUM_CE => 2,                 -- Number of clock enables (1 or 2)
   OFB_USED => FALSE,           -- Select OFB path (TRUE/FALSE)
   SERDES_MODE => "MASTER",     -- "MASTER" or "SLAVE"
   -- SRVAL_Q1 - SRVAL_Q4: Q output values when SR is used (0/1)
   SRVAL_Q1 => '0',
   SRVAL_Q2 => '0',
   SRVAL_Q3 => '0',
   SRVAL_Q4 => '0'
)
port map (
   O => O,                        -- 1-bit output: Combinatorial output
   -- Q1 - Q6: 1-bit (each) output: Registered data outputs
   Q1 => Q1,
   Q2 => Q2,
   Q3 => Q3,
   Q4 => Q4,
   Q5 => Q5,
   Q6 => Q6,
   -- SHIFTOUT1-SHIFTOUT2: 1-bit (each) output: Data width expansion output ports
   SHIFTOUT1 => SHIFTOUT1,
   SHIFTOUT2 => SHIFTOUT2,
   BITSLIP => BITSLIP,            -- 1-bit input: Bitslip enable input
   -- CE1, CE2: 1-bit (each) input: Data register clock enable inputs
   CE1 => CE1,
   CE2 => CE2,
   -- Clocks: 1-bit (each) input: ISERDESE1 clock input ports
   CLK => CLK,                    -- 1-bit input: High-speed clock input
   CLKB => CLKB,                  -- 1-bit input: High-speed secondary clock input
   CLKDIV => CLKDIV,              -- 1-bit input: Divided clock input
   OCLK => OCLK,                  -- 1-bit input: High speed output clock input used when
                                  -- INTERFACE_TYPE="MEMORY"

   -- Dynamic Clock Inversions: 1-bit (each) input: Dynamic clock inversion pins to switch clock polarity
   DYNCLKDIVSEL => DYNCLKDIVSEL, -- 1-bit input: Dynamic CLKDIV inversion input
   DYNCLKSEL => DYNCLKSEL,        -- 1-bit input: Dynamic CLK/CLKB inversion input
   -- Input Data: 1-bit (each) input: ISERDESE1 data input ports
   D => D,                        -- 1-bit input: Data input
```

```
   DDLY => DDLY,                  -- 1-bit input: Serial input data from IODELAYE1
   OFB => OFB,                    -- 1-bit input: Data feedback input from OSERDESE1
   RST => RST,                    -- 1-bit input: Active high asynchronous reset input
   -- SHIFTIN1-SHIFTIN2: 1-bit (each) input: Data width expansion input ports
   SHIFTIN1 => SHIFTIN1,
   SHIFTIN2 => SHIFTIN2
);

-- End of ISERDESE1_inst instantiation
```

Verilog Instantiation Template

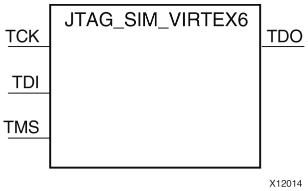
```
// ISERDESE1: Input SERial/DESerializer
//            Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

ISERDESE1 #(
   .DATA_RATE("DDR"),           // "SDR" or "DDR"
   .DATA_WIDTH(4),              // Parallel data width (2-8, 10)
   .DYN_CLKDIV_INV_EN("FALSE"), // Enable DYNCLKDIVINVSEL inversion (TRUE/FALSE)
   .DYN_CLK_INV_EN("FALSE"),    // Enable DYNCLKINVSEL inversion (TRUE/FALSE)
   // INIT_Q1 - INIT_Q4: Initial value on the Q outputs (0/1)
   .INIT_Q1(1'b0),
   .INIT_Q2(1'b0),
   .INIT_Q3(1'b0),
   .INIT_Q4(1'b0),
   .INTERFACE_TYPE("MEMORY"),   // "MEMORY", "MEMORY_DDR3", "MEMORY_QDR", "NETWORKING", or "OVERSAMPLE"
   .IOBDELAY("NONE"),           // "NONE", "IBUF", "IFD", "BOTH"
   .NUM_CE(2),                  // Number of clock enables (1 or 2)
   .OFB_USED("FALSE"),          // Select OFB path (TRUE/FALSE)
   .SERDES_MODE("MASTER"),      // "MASTER" or "SLAVE"
   // SRVAL_Q1 - SRVAL_Q4: Q output values when SR is used (0/1)
   .SRVAL_Q1(1'b0),
   .SRVAL_Q2(1'b0),
   .SRVAL_Q3(1'b0),
   .SRVAL_Q4(1'b0)
)
ISERDESE1_inst (
   .O(O),                       // 1-bit output: Combinatorial output
   // Q1 - Q6: 1-bit (each) output: Registered data outputs
   .Q1(Q1),
   .Q2(Q2),
   .Q3(Q3),
   .Q4(Q4),
   .Q5(Q5),
   .Q6(Q6),
   // SHIFTOUT1-SHIFTOUT2: 1-bit (each) output: Data width expansion output ports
   .SHIFTOUT1(SHIFTOUT1),
   .SHIFTOUT2(SHIFTOUT2),
   .BITSLIP(BITSLIP),           // 1-bit input: Bitslip enable input
   // CE1, CE2: 1-bit (each) input: Data register clock enable inputs
   .CE1(CE1),
   .CE2(CE2),
   // Clocks: 1-bit (each) input: ISERDESE1 clock input ports
   .CLK(CLK),                   // 1-bit input: High-speed clock input
   .CLKB(CLKB),                 // 1-bit input: High-speed secondary clock input
   .CLKDIV(CLKDIV),             // 1-bit input: Divided clock input
   .OCLK(OCLK),                 // 1-bit input: High speed output clock input used when
                                // INTERFACE_TYPE="MEMORY"

   // Dynamic Clock Inversions: 1-bit (each) input: Dynamic clock inversion pins to switch clock polarity
   .DYNCLKDIVSEL(DYNCLKDIVSEL), // 1-bit input: Dynamic CLKDIV inversion input
   .DYNCLKSEL(DYNCLKSEL),       // 1-bit input: Dynamic CLK/CLKB inversion input
   // Input Data: 1-bit (each) input: ISERDESE1 data input ports
   .D(D),                       // 1-bit input: Data input
   .DDLY(DDLY),                 // 1-bit input: Serial input data from IODELAYE1
   .OFB(OFB),                   // 1-bit input: Data feedback input from OSERDESE1
   .RST(RST),                   // 1-bit input: Active high asynchronous reset input
   // SHIFTIN1-SHIFTIN2: 1-bit (each) input: Data width expansion input ports
   .SHIFTIN1(SHIFTIN1),
   .SHIFTIN2(SHIFTIN2)
);

// End of ISERDESE1_inst instantiation
```

For More Information

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

JTAG_SIM_VIRTEX6

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_VIRTEX6 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 a 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.

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	"CX75T", "LX75T", "CX130T", "LX130T", "CX195T", "LX195T", "CX240T", "LX240T", "HX250T", "SX315T", "LX365T", "HX380T", "SX475T", "LX550T", "HX565T", "LX760"	"LX75T"	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_VIRTEX6: JTAG Interface Simulation Model
--                   Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

JTAG_SIM_VIRTEX6_inst : JTAG_SIM_VIRTEX6
generic map (
   PART_NAME => "LX75T") -- Specify target V6 device. Possible values are:
                         -- "CX130T","CX195T","CX240T","CX75T","HX250T",
                         -- "HX255T","HX380T","HX45T","HX565T",
                         -- "LX115T","LX130T","LX130TL","LX195T",
                         -- "LX195TL","LX240T","LX240TL","LX365T",
                         -- "LX365TL","LX40T","LX550T","LX550TL",
                         -- "LX75T","LX760","SX315T","SX475T"

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_VIRTEX6_inst instantiation
```

Verilog Instantiation Template

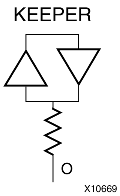
```
// JTAG_SIM_VIRTEX6: JTAG Interface Simulation Model
//                   Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

JTAG_SIM_VIRTEX6 #(
   .PART_NAME("LX75T") // Specify target V6 device.  Possible values are:
                       // "CX130T","CX195T","CX240T","CX75T","HX250T","HX255T","HX380T","HX45T","HX565T",
                    // "LX115T","LX130T","LX130TL","LX195T","LX195TL","LX240T","LX240TL","LX365T","LX365TL",
                       // "LX40T","LX550T","LX550TL","LX75T","LX760","SX315T","SX475T"
) JTAG_SIM_VIRTEX6_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_VIRTEX6_inst instantiation
```

For More Information

- See the *Synthesis and Simulation Design Guide (UG626)*.
- See the *Virtex-6 FPGA User Documentation (User Guides and Data Sheets)*.

KEEPER

Primitive: KEEPER Symbol



Introduction

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

Port Descriptions

Name	Direction	Width	Function
O	Output	1-Bit	Keeper output

Design Entry Method

Instantiation	Yes
Inference	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
--         Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

-- End of KEEPER_inst instantiation
```

Verilog Instantiation Template

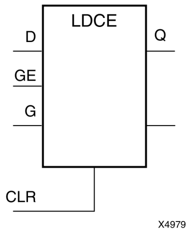
```
// KEEPER: I/O Buffer Weak Keeper
//         Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

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

// End of KEEPER_inst instantiation
```

For More Information

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

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.
--       Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

-- End of LDCE_inst instantiation
```

Verilog Instantiation Template

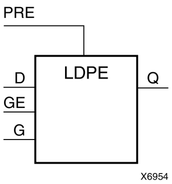
```
// LDCE:  Transparent latch with Asynchronous Reset and Gate Enable.
//       Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

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

// End of LDCE_inst instantiation
```

For More Information

See the *Virtex-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.
--        Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

-- End of LDPE_inst instantiation
```

Verilog Instantiation Template

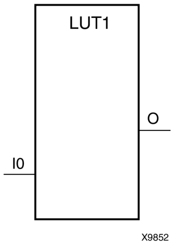
```
// LDPE:  Transparent latch with Asynchronous Preset and Gate Enable.
//        Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

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

// End of LDPE_inst instantiation
```

For More Information

See the *Virtex-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
--       Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

-- End of LUT1_inst instantiation
```

Verilog Instantiation Template

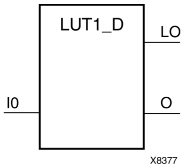
```
// LUT1: 1-input Look-Up Table with general output
//       Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

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

// End of LUT1_inst instantiation
```

For More Information

See the *Virtex-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
--         Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

-- End of LUT1_D_inst instantiation
```

Verilog Instantiation Template

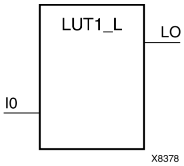
```
// LUT1_D: 1-input Look-Up Table with general and local outputs
//         Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

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

// End of LUT1_D_inst instantiation
```

For More Information

See the *Virtex-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
--         Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

-- End of LUT1_L_inst instantiation
```

Verilog Instantiation Template

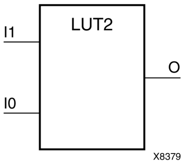
```
// LUT1_L: 1-input Look-Up Table with local output
//         Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

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

// End of LUT1_L_inst instantiation
```

For More Information

See the *Virtex-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
--       Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

-- End of LUT2_inst instantiation
```

Verilog Instantiation Template

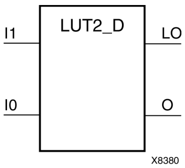
```
// LUT2: 2-input Look-Up Table with general output
//       Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

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

// End of LUT2_inst instantiation
```

For More Information

See the *Virtex-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
--         Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

-- End of LUT2_D_inst instantiation
```

Verilog Instantiation Template

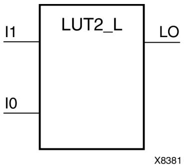
```
// LUT2_D: 2-input Look-Up Table with general and local outputs
//         Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

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

// End of LUT2_L_inst instantiation
```

For More Information

See the *Virtex-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
--         Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

-- End of LUT2_L_inst instantiation
```

Verilog Instantiation Template

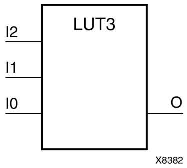
```
// LUT2_L: 2-input Look-Up Table with local output
//         Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

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

// End of LUT2_L_inst instantiation
```

For More Information

See the *Virtex-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
--       Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

-- End of LUT3_inst instantiation
```

Verilog Instantiation Template

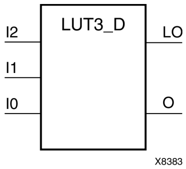
```
// LUT3: 3-input Look-Up Table with general output
//       Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

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

// End of LUT3_inst instantiation
```

For More Information

See the *Virtex-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
--         Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

-- End of LUT3_D_inst instantiation
```

Verilog Instantiation Template

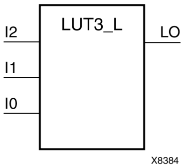
```
// LUT3_D: 3-input Look-Up Table with general and local outputs
//         Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

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

// End of LUT3_D_inst instantiation
```

For More Information

See the *Virtex-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
--         Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

-- End of LUT3_L_inst instantiation
```

Verilog Instantiation Template

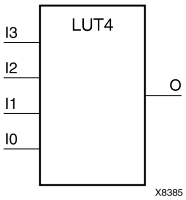
```
// LUT3_L: 3-input Look-Up Table with local output
//         Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

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

// End of LUT3_L_inst instantiation
```

For More Information

See the *Virtex-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
--       Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

-- End of LUT4_inst instantiation
```

Verilog Instantiation Template

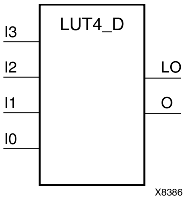
```
// LUT4: 4-input Look-Up Table with general output
//       Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

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

// End of LUT4_inst instantiation
```

For More Information

See the *Virtex-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
--         Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

-- End of LUT4_D_inst instantiation
```

Verilog Instantiation Template

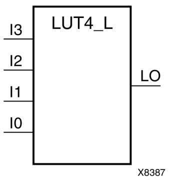
```
// LUT4_D: 4-input Look-Up Table with general and local outputs
//         Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

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

// End of LUT4_D_inst instantiation
```

For More Information

See the *Virtex-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
--         Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

-- End of LUT4_L_inst instantiation
```

Verilog Instantiation Template

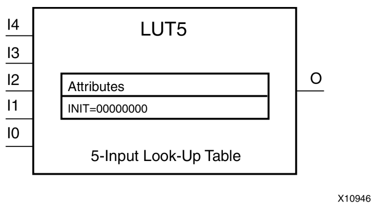
```
// LUT4_L: 4-input Look-Up Table with local output
//         Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

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

// End of LUT4_L_inst instantiation
```

For More Information

See the *Virtex-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
--       Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

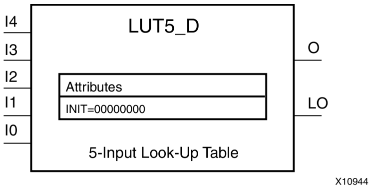
```
// LUT5: 5-input Look-Up Table with general output
//       Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

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

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
--         Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

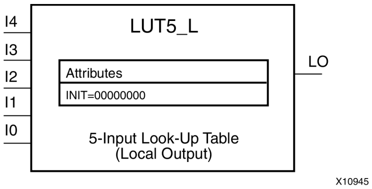
```
// LUT5_D: 5-input Look-Up Table with general and local outputs
//         Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

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

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
--         Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

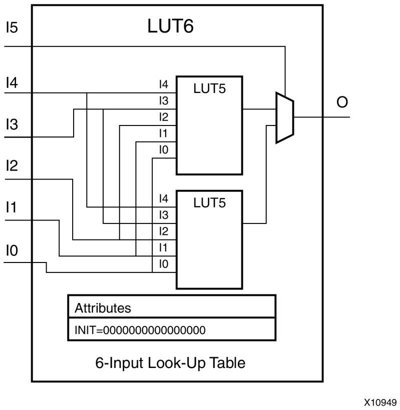
```
// LUT5_L: 5-input Look-Up Table with local output
//         Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

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

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
--       Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

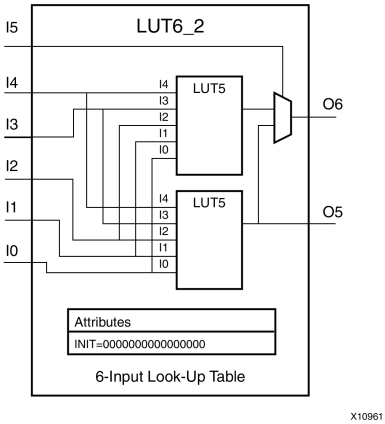
```
// LUT6: 6-input Look-Up Table with general output
//       Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

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

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
--         Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

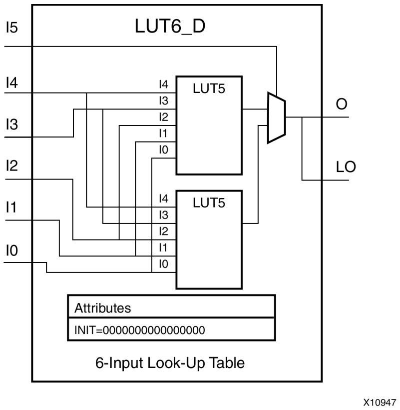
```
// LUT6_2: 6-input, 2 output Look-Up Table
//         Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

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

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
--         Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

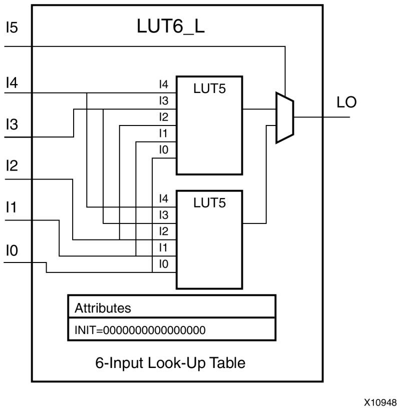
```
// LUT6_D: 6-input Look-Up Table with general and local outputs
//         Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

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

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
--         Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

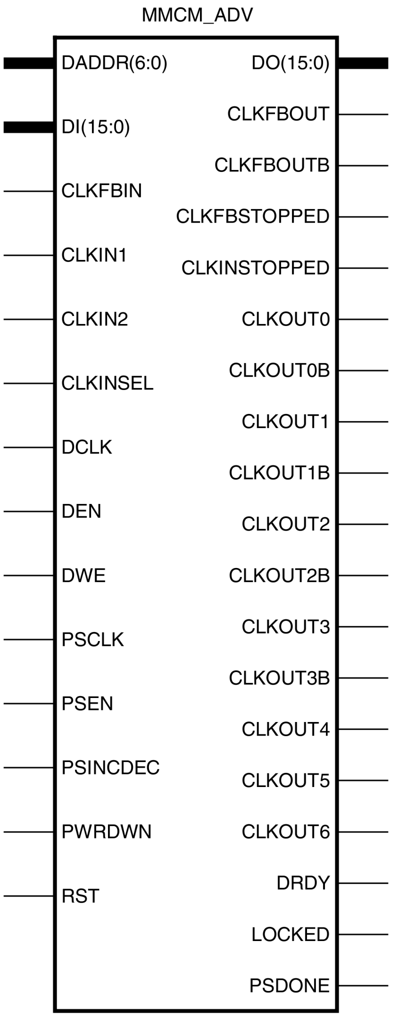
```
// LUT6_L: 6-input Look-Up Table with local output
//         Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

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

MMCM_ADV

Primitive: MMCM is a mixed signal block designed to support clock network deskew, frequency synthesis, and jitter reduction.



Introduction

The MMCM is a mixed signal block designed to support frequency synthesis, clock network deskew, and jitter reduction. The clock outputs can each have an individual divide, phase shift and duty cycle based on the same VCO frequency. Additionally, the MMCM supports dynamic phase shifting and fractional divides.

Port Descriptions

Port	Direction	Width	Function
CLKFBIN	Input	1	Feedback clock input.
CLKFBOUT	Output	1	Dedicated MMCM feedback output.
CLKFBOUTB	Output	1	Inverted CLKFBOUT.
CLKFBSTOPPED	Output	1	Status pin indicating that the feedback clock has stopped.

Port	Direction	Width	Function
CLKINSEL	Input	1	Signal controls the state of the input MUX, High = CLKIN1, Low = CLKIN2.
CLKINSTOPPED	Output	1	Status pin indicating that the input clock has stopped.
CLKIN1	Input	1	General clock input.
CLKIN2	Input	1	Secondary clock input for the MMCM reference clock.
CLKOUT[0:6]	Output	7, 1-bit	User configurable clock outputs (0 through 6) that can be divided versions of the VCO phase outputs (user controllable) from 1 (bypassed) to 128. The output clocks are phase aligned to each other (unless phase shifted) and aligned to the input clock with a proper feedback configuration.
CLKOUT[0:3]B	Output	4, 1-bit	Inverted CLKOUT[0:3].
DADDR[6:0]	Input	7	The dynamic reconfiguration address (DADDR) input bus provides a reconfiguration address for the dynamic reconfiguration. When not used, all bits must be assigned zeros.
DCLK	Input	1	The DCLK signal is the reference clock for the dynamic reconfiguration port.
DEN	Input	1	The dynamic reconfiguration enable (DEN) provides the enable control signal to access the dynamic reconfiguration feature. When the dynamic reconfiguration feature is not used, DEN must be tied Low.
DI[15:0]	Input	16	The dynamic reconfiguration data input (DI) bus provides reconfiguration data. When not used, all bits must be set to zero.
DO[15:0]	Output	16	The dynamic reconfiguration output bus provides MMCM data output when using dynamic reconfiguration.
DRDY	Output	1	The dynamic reconfiguration ready (DRDY) output provides the response to the DEN signal for the MMCMs dynamic reconfiguration feature.
DWE	Input	1	The dynamic reconfiguration write enable (DWE) input pin provides the write enable control signal to write the DI data into the DADDR address. When not used, it must be tied Low.
LOCKED	Output	1	An output from the MMCM that indicates when the MMCM has achieved phase alignment within a predefined window and frequency matching within a predefined PPM range. The MMCM automatically locks after power on. No extra reset is required. LOCKED will be deasserted if the input clock stops or the phase alignment is violated (e.g., input clock phase shift). The MMCM automatically reacquires lock after LOCKED is deasserted.
PSCLK	Input	1	Phase shift clock.
PSDONE	Output	1	Phase shift done.
PSEN	Input	1	Phase shift enable.
PSINCDEC	Input	1	Phase shift Increment/Decrement control.
PWRDWN	Input	1	Powers down instantiated but unused MMCMs.
RST	Input	1	Asynchronous reset signal.The MMCM will synchronously re-enable itself when this signal is released (i.e., MMCM re-enabled). A reset is not required when the input clock conditions change (e.g., frequency).

Design Entry Method

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

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
BANDWIDTH	String	"OPTIMIZED", "HIGH", "LOW"	"OPTIMIZED"	Specifies the MMCM programming algorithm affecting the jitter, phase margin, and other characteristics of the MMCM.
CLKFBOUT_MULT_F	3 significant digit Float	5.0 to 64.0	5.0	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 and DIVCLK_DIVIDE value, will determine the output frequency. Even though this value needs to be specified as a real number, only whole integer values are supported. For example, 6.0 is OK but 6.5 is not.
CLKFBOUT_PHASE	3 significant digit Float	-360.000 to 360.000	0.000	Specifies the phase offset in degrees of the clock feedback output. Shifting the feedback clock results in a negative phase shift of all output clocks to the MMCM.
CLKIN_PERIOD	Float (nS)	1.000 to 100.000	0.000	Specifies the input period in ns to the MMCM CLKIN1 input. Resolution is down to the ps. This information is mandatory and must be supplied.
CLKOUT0_DIVIDE_F	3 significant digit Float	1.000 to 128.000	1.000	Specifies the amount to divide the associated CLKOUT clock output if a different frequency is desired. This number in combination with the CLKFBOUT_MULT and DIVCLK_DIVIDE values will determine the output frequency.
CLKOUT[0:6]_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 CLKFBOUT_MULT and DIVCLK_DIVIDE values will determine the output frequency.
CLKOUT[0:6]_DUTY_ CYCLE	3 significant digit Float	0.001 to 0.999	0.500	Specifies the Duty Cycle of the associated CLKOUT clock output in percentage (i.e., 0.50 will generate a 50% duty cycle).
CLKOUT[0:6]_ PHASE	3 significant digit Float	-360.000 to 360.000	0.000	Specifies the phase offset in degrees of the clock feedback output. Shifting the feedback clock results in a negative phase shift of all output clocks to the MMCM.
CLKOUT4_CASCADE	Boolean	FALSE, TRUE	FALSE	Cascades the output divider (counter) into the input of the CLKOUT4 divider for an output clock divider that is greater than 128.

Attribute	Data Type	Allowed Values	Default	Description
CLOCK_HOLD	Boolean	FALSE, TRUE	FALSE	When TRUE, holds the VCO frequency close to the frequency prior to losing CLKIN.
COMPENSATION	String	"ZHOLD", "BUF_IN", "CASCADE", "EXTERNAL", "INTERNAL"	"ZHOLD"	Clock input compensation. Must be set to ZHOLD. Defines how the MMCM feedback is configured. • ZHOLD indicates the MMCM is configured to provide a negative hold time at the I/O registers. • INTERNAL indicates the MMCM is using its own internal feedback path so no delay is being compensated. • EXTERNAL indicates a network external to the FPGA is being compensated. • CASCADE indicates cascading of 2 MMCM. • BUF_IN indicates that the configuration does not match with the other compensation modes and no delay will be compensated. This is the case if a clock input is driven by a BUFG/BUFH/BUFR/GT.
DIVCLK_DIVIDE	Integer	1 to 128	1	Specifies the division ratio for all output clocks with respect to the input clock. Effectively divides the CLKIN going into the PFD.
REF_JITTER1	3 significant digit Float	0.000 to 0.999	0.010	Allows specification of the expected jitter on CLKIN1 in order to better optimize MMCM performance. A bandwidth setting of OPTIMIZED will attempt to choose the best parameter for input clocking when unknown. If known, then the value provided should be specified in terms of the UI percentage (the maximum peak to peak value) of the expected jitter on the input clock.
REF_JITTER2	3 significant digit Float	0.000 to 0.999	0.010	Allows specification of the expected jitter on CLKIN2 in order to better optimize MMCM performance. A bandwidth setting of OPTIMIZED will attempt to choose the best parameter for input clocking when unknown. If known, then the value provided should be specified in terms of the UI percentage (the maximum peak to peak value) of the expected jitter on the input clock.
STARTUP_WAIT	Boolean	FALSE	FALSE	This attribute is not supported.
CLKFBOUT_USE_ FINE_PS	Boolean	FALSE, TRUE	FALSE	CLKFBOUT Counter variable fine phase shift enable
CLKOUT[0:6]_USE_ FINE_PS	Boolean	FALSE, TRUE	FALSE	CLKOUT[1:6] variable fine phase shift enable.

VHDL Instantiation Template

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

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


-- MMCM_ADV: Advanced Mixed Mode Clock Manager
--           Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

MMCM_ADV_inst : MMCM_ADV
generic map (
   BANDWIDTH => "OPTIMIZED",      -- Jitter programming ("HIGH","LOW","OPTIMIZED")
   CLKFBOUT_MULT_F => 5.0,        -- Multiply value for all CLKOUT (5.0-64.0).
   CLKFBOUT_PHASE => 0.0,         -- Phase offset in degrees of CLKFB (0.00-360.00).
   -- CLKIN_PERIOD: Input clock period in ns to ps resolution (i.e. 33.333 is 30 MHz).
   CLKIN1_PERIOD => 0.0,
   CLKIN2_PERIOD => 0.0,
   CLKOUT0_DIVIDE_F => 1.0,       -- Divide amount for CLKOUT0 (1.000-128.000).
   -- CLKOUT0_DUTY_CYCLE - CLKOUT6_DUTY_CYCLE: Duty cycle for CLKOUT outputs (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,
   CLKOUT6_DUTY_CYCLE => 0.5,
   -- CLKOUT0_PHASE - CLKOUT6_PHASE: Phase offset for CLKOUT outputs (-360.000-360.000).
   CLKOUT0_PHASE => 0.0,
   CLKOUT1_PHASE => 0.0,
   CLKOUT2_PHASE => 0.0,
   CLKOUT3_PHASE => 0.0,
   CLKOUT4_PHASE => 0.0,
   CLKOUT5_PHASE => 0.0,
   CLKOUT6_PHASE => 0.0,
   -- CLKOUT1_DIVIDE - CLKOUT6_DIVIDE: Divide amount for CLKOUT (1-128)
   CLKOUT1_DIVIDE => 1,
   CLKOUT2_DIVIDE => 1,
   CLKOUT3_DIVIDE => 1,
   CLKOUT4_DIVIDE => 1,
   CLKOUT5_DIVIDE => 1,
   CLKOUT6_DIVIDE => 1,
   CLKOUT4_CASCADE => FALSE,      -- Cascase CLKOUT4 counter with CLKOUT6 (TRUE/FALSE)
   CLOCK_HOLD => FALSE,           -- Hold VCO Frequency (TRUE/FALSE)
   COMPENSATION => "ZHOLD",       -- "ZHOLD", "INTERNAL", "EXTERNAL", "CASCADE" or "BUF_IN"
   DIVCLK_DIVIDE => 1,            -- Master division value (1-80)
   -- REF_JITTER: Reference input jitter in UI (0.000-0.999).
   REF_JITTER1 => 0.0,
   REF_JITTER2 => 0.0,
   STARTUP_WAIT => FALSE,         -- Not supported. Must be set to FALSE.
   -- USE_FINE_PS: Fine phase shift enable (TRUE/FALSE)
   CLKFBOUT_USE_FINE_PS => FALSE,
   CLKOUT0_USE_FINE_PS => FALSE,
   CLKOUT1_USE_FINE_PS => FALSE,
   CLKOUT2_USE_FINE_PS => FALSE,
   CLKOUT3_USE_FINE_PS => FALSE,
   CLKOUT4_USE_FINE_PS => FALSE,
   CLKOUT5_USE_FINE_PS => FALSE,
   CLKOUT6_USE_FINE_PS => FALSE
)
port map (
   -- Clock Outputs: 1-bit (each) output: User configurable clock outputs
   CLKOUT0 => CLKOUT0,           -- 1-bit output: CLKOUT0 output
   CLKOUT0B => CLKOUT0B,         -- 1-bit output: Inverted CLKOUT0 output
   CLKOUT1 => CLKOUT1,           -- 1-bit output: CLKOUT1 output
   CLKOUT1B => CLKOUT1B,         -- 1-bit output: Inverted CLKOUT1 output
   CLKOUT2 => CLKOUT2,           -- 1-bit output: CLKOUT2 output
   CLKOUT2B => CLKOUT2B,         -- 1-bit output: Inverted CLKOUT2 output
   CLKOUT3 => CLKOUT3,           -- 1-bit output: CLKOUT3 output
   CLKOUT3B => CLKOUT3B,         -- 1-bit output: Inverted CLKOUT3 output
   CLKOUT4 => CLKOUT4,           -- 1-bit output: CLKOUT4 output
```

```
   CLKOUT5 => CLKOUT5,           -- 1-bit output: CLKOUT5 output
   CLKOUT6 => CLKOUT6,           -- 1-bit output: CLKOUT6 output
   -- DRP Ports: 16-bit (each) output: Dynamic reconfiguration ports
   DO => DO,                     -- 16-bit output: DRP data output
   DRDY => DRDY,                 -- 1-bit output: DRP ready output
   -- Dynamic Phase Shift Ports: 1-bit (each) output: Ports used for dynamic phase shifting of the outputs
   PSDONE => PSDONE,             -- 1-bit output: Phase shift done output
   -- Feedback Clocks: 1-bit (each) output: Clock feedback ports
   CLKFBOUT => CLKFBOUT,         -- 1-bit output: Feedback clock output
   CLKFBOUTB => CLKFBOUTB,       -- 1-bit output: Inverted CLKFBOUT
   -- Status Ports: 1-bit (each) output: MMCM status ports
   CLKFBSTOPPED => CLKFBSTOPPED, -- 1-bit output: Feedback clock stopped output
   CLKINSTOPPED => CLKINSTOPPED, -- 1-bit output: Input clock stopped output
   LOCKED => LOCKED,             -- 1-bit output: LOCK output
   -- Clock Inputs: 1-bit (each) input: Clock inputs
   CLKIN1 => CLKIN1,             -- 1-bit input: Primary clock input
   CLKIN2 => CLKIN2,             -- 1-bit input: Secondary clock input
   -- Control Ports: 1-bit (each) input: MMCM control ports
   CLKINSEL => CLKINSEL,         -- 1-bit input: Clock select input
   PWRDWN => PWRDWN,             -- 1-bit input: Power-down input
   RST => RST,                   -- 1-bit input: Reset input
   -- DRP Ports: 7-bit (each) input: Dynamic reconfiguration ports
   DADDR => DADDR,               -- 7-bit input: DRP adrress input
   DCLK => DCLK,                 -- 1-bit input: DRP clock input
   DEN => DEN,                   -- 1-bit input: DRP enable input
   DI => DI,                     -- 16-bit input: DRP data input
   DWE => DWE,                   -- 1-bit input: DRP write enable input
   -- Dynamic Phase Shift Ports: 1-bit (each) input: Ports used for dynamic phase shifting of the outputs
   PSCLK => PSCLK,               -- 1-bit input: Phase shift clock input
   PSEN => PSEN,                 -- 1-bit input: Phase shift enable input
   PSINCDEC => PSINCDEC,         -- 1-bit input: Phase shift increment/decrement input
   -- Feedback Clocks: 1-bit (each) input: Clock feedback ports
   CLKFBIN => CLKFBIN            -- 1-bit input: Feedback clock input
);

-- End of MMCM_ADV_inst instantiation
```

Verilog Instantiation Template

```
// MMCM_ADV: Advanced Mixed Mode Clock Manager
//           Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

MMCM_ADV #(
   .BANDWIDTH("OPTIMIZED"),        // Jitter programming ("HIGH","LOW","OPTIMIZED")
   .CLKFBOUT_MULT_F(5.0),          // Multiply value for all CLKOUT (5.0-64.0).
   .CLKFBOUT_PHASE(0.0),           // Phase offset in degrees of CLKFB (0.00-360.00).
   // CLKIN_PERIOD: Input clock period in ns to ps resolution (i.e. 33.333 is 30 MHz).
   .CLKIN1_PERIOD(0.0),
   .CLKIN2_PERIOD(0.0),
   .CLKOUT0_DIVIDE_F(1.0),         // Divide amount for CLKOUT0 (1.000-128.000).
   // CLKOUT0_DUTY_CYCLE - CLKOUT6_DUTY_CYCLE: Duty cycle for CLKOUT outputs (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),
   .CLKOUT6_DUTY_CYCLE(0.5),
   // CLKOUT0_PHASE - CLKOUT6_PHASE: Phase offset for CLKOUT outputs (-360.000-360.000).
   .CLKOUT0_PHASE(0.0),
   .CLKOUT1_PHASE(0.0),
   .CLKOUT2_PHASE(0.0),
   .CLKOUT3_PHASE(0.0),
   .CLKOUT4_PHASE(0.0),
   .CLKOUT5_PHASE(0.0),
   .CLKOUT6_PHASE(0.0),
   // CLKOUT1_DIVIDE - CLKOUT6_DIVIDE: Divide amount for CLKOUT (1-128)
   .CLKOUT1_DIVIDE(1),
   .CLKOUT2_DIVIDE(1),
   .CLKOUT3_DIVIDE(1),
```

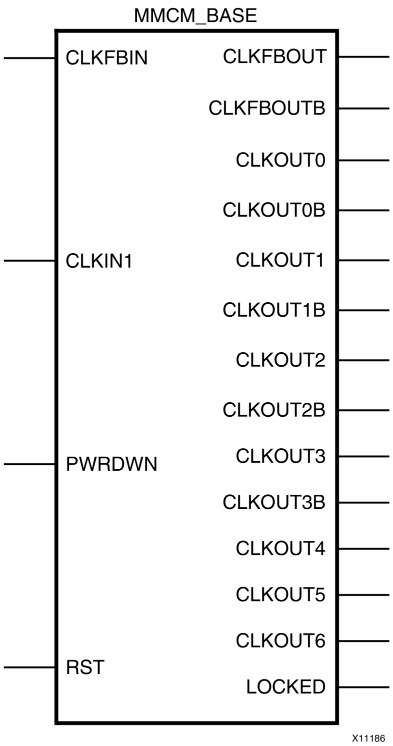
```
   .CLKOUT4_DIVIDE(1),
   .CLKOUT5_DIVIDE(1),
   .CLKOUT6_DIVIDE(1),
   .CLKOUT4_CASCADE("FALSE"),      // Cascase CLKOUT4 counter with CLKOUT6 (TRUE/FALSE)
   .CLOCK_HOLD("FALSE"),           // Hold VCO Frequency (TRUE/FALSE)
   .COMPENSATION("ZHOLD"),         // "ZHOLD", "INTERNAL", "EXTERNAL", "CASCADE" or "BUF_IN"
   .DIVCLK_DIVIDE(1),              // Master division value (1-80)
   // REF_JITTER: Reference input jitter in UI (0.000-0.999).
   .REF_JITTER1(0.0),
   .REF_JITTER2(0.0),
   .STARTUP_WAIT("FALSE"),         // Not supported. Must be set to FALSE.
   // USE_FINE_PS: Fine phase shift enable (TRUE/FALSE)
   .CLKFBOUT_USE_FINE_PS("FALSE"),
   .CLKOUT0_USE_FINE_PS("FALSE"),
   .CLKOUT1_USE_FINE_PS("FALSE"),
   .CLKOUT2_USE_FINE_PS("FALSE"),
   .CLKOUT3_USE_FINE_PS("FALSE"),
   .CLKOUT4_USE_FINE_PS("FALSE"),
   .CLKOUT5_USE_FINE_PS("FALSE"),
   .CLKOUT6_USE_FINE_PS("FALSE")
)
MMCM_ADV_inst (
   // Clock Outputs: 1-bit (each) output: User configurable clock outputs
   .CLKOUT0(CLKOUT0),           // 1-bit output: CLKOUT0 output
   .CLKOUT0B(CLKOUT0B),         // 1-bit output: Inverted CLKOUT0 output
   .CLKOUT1(CLKOUT1),           // 1-bit output: CLKOUT1 output
   .CLKOUT1B(CLKOUT1B),         // 1-bit output: Inverted CLKOUT1 output
   .CLKOUT2(CLKOUT2),           // 1-bit output: CLKOUT2 output
   .CLKOUT2B(CLKOUT2B),         // 1-bit output: Inverted CLKOUT2 output
   .CLKOUT3(CLKOUT3),           // 1-bit output: CLKOUT3 output
   .CLKOUT3B(CLKOUT3B),         // 1-bit output: Inverted CLKOUT3 output
   .CLKOUT4(CLKOUT4),           // 1-bit output: CLKOUT4 output
   .CLKOUT5(CLKOUT5),           // 1-bit output: CLKOUT5 output
   .CLKOUT6(CLKOUT6),           // 1-bit output: CLKOUT6 output
   // DRP Ports: 16-bit (each) output: Dynamic reconfiguration ports
   .DO(DO),                     // 16-bit output: DRP data output
   .DRDY(DRDY),                 // 1-bit output: DRP ready output
   // Dynamic Phase Shift Ports: 1-bit (each) output: Ports used for dynamic phase shifting of the outputs
   .PSDONE(PSDONE),             // 1-bit output: Phase shift done output
   // Feedback Clocks: 1-bit (each) output: Clock feedback ports
   .CLKFBOUT(CLKFBOUT),         // 1-bit output: Feedback clock output
   .CLKFBOUTB(CLKFBOUTB),       // 1-bit output: Inverted CLKFBOUT
   // Status Ports: 1-bit (each) output: MMCM status ports
   .CLKFBSTOPPED(CLKFBSTOPPED), // 1-bit output: Feedback clock stopped output
   .CLKINSTOPPED(CLKINSTOPPED), // 1-bit output: Input clock stopped output
   .LOCKED(LOCKED),             // 1-bit output: LOCK output
   // Clock Inputs: 1-bit (each) input: Clock inputs
   .CLKIN1(CLKIN1),             // 1-bit input: Primary clock input
   .CLKIN2(CLKIN2),             // 1-bit input: Secondary clock input
   // Control Ports: 1-bit (each) input: MMCM control ports
   .CLKINSEL(CLKINSEL),         // 1-bit input: Clock select input
   .PWRDWN(PWRDWN),             // 1-bit input: Power-down input
   .RST(RST),                   // 1-bit input: Reset input
   // DRP Ports: 7-bit (each) input: Dynamic reconfigration ports
   .DADDR(DADDR),               // 7-bit input: DRP adrress input
   .DCLK(DCLK),                 // 1-bit input: DRP clock input
   .DEN(DEN),                   // 1-bit input: DRP enable input
   .DI(DI),                     // 16-bit input: DRP data input
   .DWE(DWE),                   // 1-bit input: DRP write enable input
   // Dynamic Phase Shift Ports: 1-bit (each) input: Ports used for dynamic phase shifting of the outputs
   .PSCLK(PSCLK),               // 1-bit input: Phase shift clock input
   .PSEN(PSEN),                 // 1-bit input: Phase shift enable input
   .PSINCDEC(PSINCDEC),         // 1-bit input: Phase shift increment/decrement input
   // Feedback Clocks: 1-bit (each) input: Clock feedback ports
   .CLKFBIN(CLKFBIN)            // 1-bit input: Feedback clock input
);

// End of MMCM_ADV_inst instantiation
```

For More Information

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

MMCM_BASE

Primitive: Mixed signal block designed to support clock network deskew, frequency synthesis, and jitter reduction.



Introduction

This component is a mixed signal block designed to support clock network deskew, frequency synthesis, and jitter reduction. The seven "O" counters can be independently programmed which means O0 could be programmed to do a divide by 2 while O1 is programmed to do a divide by 3. The only constraint is that the VCO operating frequency must be the same for all the output counters since a single VCO drives all the counters. The CLKFBOUT and CLKFBOUTB pins can be used to drive logic but it must be equal to the CLKin frequency.

Port Descriptions

Port	Direction	Width	Function
CLKFBIN	Input	1	Feedback clock input.
CLKFBOUT	Output	1	Dedicated MMCM feedback output.
CLKFBOUTB	Output	1	Inverted MMCM feedback clock output.
CLKIN1	Input	1	General clock input.
CLKOUT[0:6]	Output	7, 1-bit	User configurable clock outputs (0 through 6) that can be divided versions of the VCO phase outputs (user controllable) from 1 (bypassed) to 128. The output clocks are phase aligned to each other (unless phase shifted) and aligned to the input clock with a proper feedback configuration.
CLKOUT[0:3]B	Output	4, 1-bit	Inverted CLKOUT[0:3].

Port	Direction	Width	Function
LOCKED	Output	1	An output from the MMCM that indicates when the MMCM has achieved phase alignment within a predefined window and frequency matching within a predefined PPM range. The MMCM automatically locks after power on. No extra reset is required. LOCKED will be deasserted if the input clock stops or the phase alignment is violated (e.g., input clock phase shift). The MMCM automatically reacquires lock after LOCKED is deasserted.
PWRDWN	Input	1	Powers down instantiated but unused MMCMs.
RST	Input	1	Asynchronous reset signal. The MMCM will synchronously re-enable itself when this signal is released (i.e., MMCM re-enabled). A reset is not required when the input clock conditions change (e.g., frequency).

Design Entry Method

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

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
BANDWIDTH	String	"OPTIMIZED", "HIGH", "LOW"	"OPTIMIZED"	Specifies the MMCM programming algorithm affecting the jitter, phase margin, and other characteristics of the MMCM.
CLKFBOUT_MULT_F	3 significant digit Float	5.0 to 64.0	5.0	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 and DIVCLK_DIVIDE value, will determine the output frequency.
CLKFBOUT_PHASE	3 significant digit Float	-360.000 to 360.000	0.000	Specifies the phase offset in degrees of the clock feedback output. Shifting the feedback clock results in a negative phase shift of all output clocks to the MMCM.
CLKIN1_PERIOD	Float (nS)	1.000 to 1000.000	0.000	Specifies the input period in ns to the MMCM CLKIN1 input. Resolution is down to the ps. This information is mandatory and must be supplied.
CLKOUT0_DIVIDE_F	3 significant digit Float	1.000 to 128.000	1.000	Specifies the amount to divide the associated CLKOUT clock output if a different frequency is desired. This number in combination with the CLKFBOUT_MULT and DIVCLK_DIVIDE values will determine the output frequency.
CLKOUT[0:6]_DUTY_ CYCLE	3 significant digit Float	0.001 to 0.999	0.500	Specifies the Duty Cycle of the associated CLKOUT clock output in percentage (for instance, 0.50 will generate a 50% duty cycle).

Attribute	Data Type	Allowed Values	Default	Description
CLKOUT[0:6]_PHASE	3 significant digit Float	-360.000 to 360.000	0.000	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).
CLOCK_HOLD	Boolean	FALSE, TRUE	FALSE	When TRUE, holds the VCO frequency close to the frequency prior to losing CLKIN.
DIVCLK_DIVIDE	Integer	1 to 128	1	Specifies the division ratio for all output clocks with respect to the input clock. Effectively divides the CLKIN going into the PFD.
REF_JITTER1	3 significant digit Float	0.000 to 0.999	0.010	Allows specification of the expected jitter on the reference clock in order to better optimize MMCM performance. A bandwidth setting of OPTIMIZED will attempt to choose the best parameter for input clocking when unknown. If known, then the value provided should be specified in terms of the UI percentage (the maximum peak to peak value) of the expected jitter on the input clock.
STARTUP_WAIT	Boolean	FALSE	FALSE	This attribute is not supported.

VHDL Instantiation Template

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

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


-- MMCM_BASE: Base Mixed Mode Clock Manager
--            Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

MMCM_BASE_inst : MMCM_BASE
generic map (
   BANDWIDTH => "OPTIMIZED",  -- Jitter programming ("HIGH","LOW","OPTIMIZED")
   CLKFBOUT_MULT_F => 5.0,    -- Multiply value for all CLKOUT (5.0-64.0).
   CLKFBOUT_PHASE => 0.0,     -- Phase offset in degrees of CLKFB (0.00-360.00).
   CLKIN1_PERIOD => 0.0,      -- Input clock period in ns to ps resolution (i.e. 33.333 is 30 MHz).
   CLKOUT0_DIVIDE_F => 1.0,   -- Divide amount for CLKOUT0 (1.000-128.000).
   -- CLKOUT0_DUTY_CYCLE - CLKOUT6_DUTY_CYCLE: Duty cycle for each CLKOUT (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,
   CLKOUT6_DUTY_CYCLE => 0.5,
   -- CLKOUT0_PHASE - CLKOUT6_PHASE: Phase offset for each CLKOUT (-360.000-360.000).
   CLKOUT0_PHASE => 0.0,
   CLKOUT1_PHASE => 0.0,
   CLKOUT2_PHASE => 0.0,
   CLKOUT3_PHASE => 0.0,
   CLKOUT4_PHASE => 0.0,
   CLKOUT5_PHASE => 0.0,
   CLKOUT6_PHASE => 0.0,
   -- CLKOUT1_DIVIDE - CLKOUT6_DIVIDE: Divide amount for each CLKOUT (1-128)
   CLKOUT1_DIVIDE => 1,
   CLKOUT2_DIVIDE => 1,
```

```
   CLKOUT3_DIVIDE => 1,
   CLKOUT4_DIVIDE => 1,
   CLKOUT5_DIVIDE => 1,
   CLKOUT6_DIVIDE => 1,
   CLKOUT4_CASCADE => FALSE,  -- Cascase CLKOUT4 counter with CLKOUT6 (TRUE/FALSE)
   CLOCK_HOLD => FALSE,       -- Hold VCO Frequency (TRUE/FALSE)
   DIVCLK_DIVIDE => 1,        -- Master division value (1-80)
   REF_JITTER1 => 0.0,        -- Reference input jitter in UI (0.000-0.999).
   STARTUP_WAIT => FALSE      -- Not supported. Must be set to FALSE.
)
port map (
   -- Clock Outputs: 1-bit (each) output: User configurable clock outputs
   CLKOUT0 => CLKOUT0,     -- 1-bit output: CLKOUT0 output
   CLKOUT0B => CLKOUT0B,   -- 1-bit output: Inverted CLKOUT0 output
   CLKOUT1 => CLKOUT1,     -- 1-bit output: CLKOUT1 output
   CLKOUT1B => CLKOUT1B,   -- 1-bit output: Inverted CLKOUT1 output
   CLKOUT2 => CLKOUT2,     -- 1-bit output: CLKOUT2 output
   CLKOUT2B => CLKOUT2B,   -- 1-bit output: Inverted CLKOUT2 output
   CLKOUT3 => CLKOUT3,     -- 1-bit output: CLKOUT3 output
   CLKOUT3B => CLKOUT3B,   -- 1-bit output: Inverted CLKOUT3 output
   CLKOUT4 => CLKOUT4,     -- 1-bit output: CLKOUT4 output
   CLKOUT5 => CLKOUT5,     -- 1-bit output: CLKOUT5 output
   CLKOUT6 => CLKOUT6,     -- 1-bit output: CLKOUT6 output
   -- Feedback Clocks: 1-bit (each) output: Clock feedback ports
   CLKFBOUT => CLKFBOUT,   -- 1-bit output: Feedback clock output
   CLKFBOUTB => CLKFBOUTB, -- 1-bit output: Inverted CLKFBOUT output
   -- Status Port: 1-bit (each) output: MMCM status ports
   LOCKED => LOCKED,       -- 1-bit output: LOCK output
   -- Clock Input: 1-bit (each) input: Clock input
   CLKIN1 => CLKIN1,
   -- Control Ports: 1-bit (each) input: MMCM control ports
   PWRDWN => PWRDWN,       -- 1-bit input: Power-down input
   RST => RST,             -- 1-bit input: Reset input
   -- Feedback Clocks: 1-bit (each) input: Clock feedback ports
   CLKFBIN => CLKFBIN      -- 1-bit input: Feedback clock input
);

-- End of MMCM_BASE_inst instantiation
```

Verilog Instantiation Template

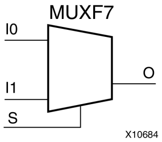
```
// MMCM_BASE: Base Mixed Mode Clock Manager
//            Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

MMCM_BASE #(
   .BANDWIDTH("OPTIMIZED"),   // Jitter programming ("HIGH","LOW","OPTIMIZED")
   .CLKFBOUT_MULT_F(5.0),     // Multiply value for all CLKOUT (5.0-64.0).
   .CLKFBOUT_PHASE(0.0),      // Phase offset in degrees of CLKFB (0.00-360.00).
   .CLKIN1_PERIOD(0.0),       // Input clock period in ns to ps resolution (i.e. 33.333 is 30 MHz).
   .CLKOUT0_DIVIDE_F(1.0),    // Divide amount for CLKOUT0 (1.000-128.000).
   // CLKOUT0_DUTY_CYCLE - CLKOUT6_DUTY_CYCLE: Duty cycle for each CLKOUT (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),
   .CLKOUT6_DUTY_CYCLE(0.5),
   // CLKOUT0_PHASE - CLKOUT6_PHASE: Phase offset for each CLKOUT (-360.000-360.000).
   .CLKOUT0_PHASE(0.0),
   .CLKOUT1_PHASE(0.0),
   .CLKOUT2_PHASE(0.0),
   .CLKOUT3_PHASE(0.0),
   .CLKOUT4_PHASE(0.0),
   .CLKOUT5_PHASE(0.0),
   .CLKOUT6_PHASE(0.0),
   // CLKOUT1_DIVIDE - CLKOUT6_DIVIDE: Divide amount for each CLKOUT (1-128)
   .CLKOUT1_DIVIDE(1),
   .CLKOUT2_DIVIDE(1),
   .CLKOUT3_DIVIDE(1),
   .CLKOUT4_DIVIDE(1),
   .CLKOUT5_DIVIDE(1),
   .CLKOUT6_DIVIDE(1),
   .CLKOUT4_CASCADE("FALSE"), // Cascase CLKOUT4 counter with CLKOUT6 (TRUE/FALSE)
   .CLOCK_HOLD("FALSE"),      // Hold VCO Frequency (TRUE/FALSE)
   .DIVCLK_DIVIDE(1),         // Master division value (1-80)
   .REF_JITTER1(0.0),         // Reference input jitter in UI (0.000-0.999).
   .STARTUP_WAIT("FALSE")     // Not supported. Must be set to FALSE.
)
MMCM_BASE_inst (
   // Clock Outputs: 1-bit (each) output: User configurable clock outputs
   .CLKOUT0(CLKOUT0),     // 1-bit output: CLKOUT0 output
   .CLKOUT0B(CLKOUT0B),   // 1-bit output: Inverted CLKOUT0 output
   .CLKOUT1(CLKOUT1),     // 1-bit output: CLKOUT1 output
   .CLKOUT1B(CLKOUT1B),   // 1-bit output: Inverted CLKOUT1 output
   .CLKOUT2(CLKOUT2),     // 1-bit output: CLKOUT2 output
   .CLKOUT2B(CLKOUT2B),   // 1-bit output: Inverted CLKOUT2 output
   .CLKOUT3(CLKOUT3),     // 1-bit output: CLKOUT3 output
   .CLKOUT3B(CLKOUT3B),   // 1-bit output: Inverted CLKOUT3 output
   .CLKOUT4(CLKOUT4),     // 1-bit output: CLKOUT4 output
   .CLKOUT5(CLKOUT5),     // 1-bit output: CLKOUT5 output
   .CLKOUT6(CLKOUT6),     // 1-bit output: CLKOUT6 output
   // Feedback Clocks: 1-bit (each) output: Clock feedback ports
   .CLKFBOUT(CLKFBOUT),   // 1-bit output: Feedback clock output
   .CLKFBOUTB(CLKFBOUTB), // 1-bit output: Inverted CLKFBOUT output
   // Status Port: 1-bit (each) output: MMCM status ports
   .LOCKED(LOCKED),       // 1-bit output: LOCK output
   // Clock Input: 1-bit (each) input: Clock input
   .CLKIN1(CLKIN1),
   // Control Ports: 1-bit (each) input: MMCM control ports
   .PWRDWN(PWRDWN),       // 1-bit input: Power-down input
   .RST(RST),             // 1-bit input: Reset input
   // Feedback Clocks: 1-bit (each) input: Clock feedback ports
   .CLKFBIN(CLKFBIN)      // 1-bit input: Feedback clock input
);

// End of MMCM_BASE_inst instantiation
```

For More Information

See the *Virtex-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
--        Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

-- End of MUXF7_inst instantiation
```

Verilog Instantiation Template

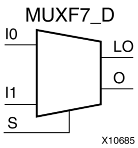
```
// MUXF7: CLB MUX to tie two LUT6's together with general output
//        Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

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

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
--          Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

MUXF7_D_inst : MUXF7_D
port map (
   LO => LO,  -- Output 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

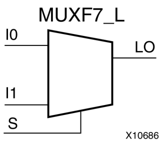
```
// MUXF7_D: CLB MUX to tie two LUT6's together with general and local outputs
//          Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

MUXF7_D MUXF7_D_inst (
   .LO(LO),  // Output 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 *Virtex-6 FPGA User Documentation (User Guides and Data Sheets)*.

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
--          Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

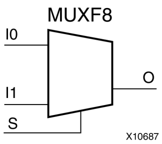
```
// MUXF7_L: CLB MUX to tie two LUT6's together with local output
//          Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

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

MUXF8

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



Introduction

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

Logic Table

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

Port Descriptions

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
--        Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

-- End of MUXF8_inst instantiation
```

Verilog Instantiation Template

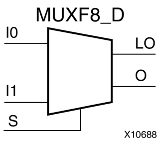
```
// MUXF8: CLB MUX to tie two MUXF7's together with general output
//        Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

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

// End of MUXF8_inst instantiation
```

For More Information

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

MUXF8_D

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



Introduction

This design element provides a multiplexer function in eight slices for creating a function-of-8 look-up table or a 32-to-1 multiplexer in combination with the associated 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
--          Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

MUXF8_D_inst : MUXF8_D
port map (
   LO => LO,  -- Output 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

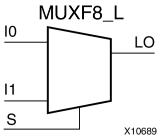
```
// MUXF8_D: CLB MUX to tie two MUXF7's together with general and local outputs
//          Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

MUXF8_D MUXF8_D_inst (
   .LO(LO),   // Output 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 *Virtex-6 FPGA User Documentation (User Guides and Data Sheets)*.

MUXF8_L

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



Introduction

This design element provides a multiplexer function in eight slices for creating a function-of-8 look-up table or a 32-to-1 multiplexer in combination with the associated 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
--          Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

-- End of MUXF8_L_inst instantiation
```

Verilog Instantiation Template

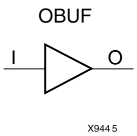
```
// MUXF8_L: CLB MUX to tie two MUXF7's together with local output
//          Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

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

// End of MUXF8_L_inst instantiation
```

For More Information

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

OBUF

Primitive: Output Buffer



Introduction

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

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

Port Descriptions

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

Design Entry Method

Instantiation	Yes
Inference	Recommended
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. It is suggested that you set this to the lowest setting tolerable for the design drive and timing requirements.
IOSTANDARD	String	See Data Sheet	"DEFAULT"	Assigns an I/O standard to the element.
SLEW	String	"SLOW" or "FAST"	"SLOW"	Specifies the slew rate of the output driver. Consult the product Data Sheet for recommendations of the best setting for this attribute.

VHDL Instantiation Template

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

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


-- OBUF: Single-ended Output Buffer
--       Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

-- End of OBUF_inst instantiation
```

Verilog Instantiation Template

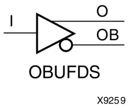
```
// OBUF: Single-ended Output Buffer
//       Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

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

// End of OBUF_inst instantiation
```

For More Information

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

OBUFDS

Primitive: Differential Signaling Output Buffer



Introduction

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

Logic Table

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

Port Descriptions

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
--         Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

-- End of OBUFDS_inst instantiation
```

Verilog Instantiation Template

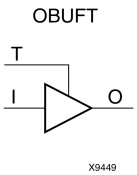
```
// OBUFDS: Differential Output Buffer
//         Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

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

// End of OBUFDS_inst instantiation
```

For More Information

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

OBUFT

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



Introduction

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

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

Logic Table

Inputs		Outputs
T	**I**	**O**
1	X	Z
0	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
--        Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

-- End of OBUFT_inst instantiation
```

Verilog Instantiation Template

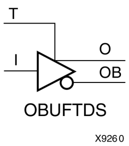
```
// OBUFT: Single-ended 3-state Output Buffer
//        All devices
// Xilinx HDL Libraries Guide, version 14.2

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

  // End of OBUFT_inst instantiation
```

For More Information

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

OBUFTDS

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



Introduction

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

Logic Table

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

Port Descriptions

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
--          Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

-- End of OBUFTDS_inst instantiation
```

Verilog Instantiation Template

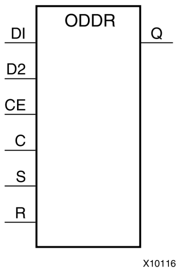
```
// OBUFTDS: Differential 3-state Output Buffer
//          Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

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

// End of OBUFTDS_inst instantiation
```

For More Information

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

ODDR

Primitive: Dedicated Dual Data Rate (DDR) Output Register



Introduction

This design element is a dedicated output register for use in transmitting dual data rate (DDR) signals from FPGA devices. The ODDR interface with the FPGA fabric is not limited to opposite clock edges. The ODDR is available with modes that allow data to be presented from the FPGA fabric at the same clock edge. This feature allows designers to avoid additional timing complexities and CLB usage. In addition, the ODDR works in conjunction with SelectIO™ features.

ODDR Modes

This element has two modes of operation. These modes are set by the DDR_CLK_EDGE attribute.

- **OPPOSITE_EDGE mode -** The data transmit interface uses the classic DDR methodology. Given a data and clock at pin D1-2 and C respectively, D1 is sampled at every positive edge of clock C, and D2 is sampled at every negative edge of clock C. Q changes every clock edge.
- **SAME_EDGE mode -** Data is still transmitted at the output of the ODDR by opposite edges of clock C. However, the two inputs to the ODDR are clocked with a positive clock edge of clock signal C and an extra register is clocked with a negative clock edge of clock signal C. Using this feature, DDR data can now be presented into the ODDR at the same clock edge.

Port Descriptions

Port	Direction	Width	Function
Q	Output	1	Data Output (DDR) - The ODDR output that connects to the IOB pad.
C	Input	1	Clock Input - The C pin represents the clock input pin.
CE	Input	1	Clock Enable Input - When asserted High, this port enables the clock input on port C.
D1 : D2	Input	1 (each)	Data Input - This pin is where the DDR data is presented into the ODDR module.
R	Input	1	Reset - Depends on how SRTYPE is set.
S	Input	1	Set - Active High asynchronous set pin. This pin can also be Synchronous depending on the SRTYPE attribute.

Design Entry Method

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

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
DDR_CLK_ EDGE	String	"OPPOSITE_EDGE", "SAME_EDGE"	"OPPOSITE_ EDGE"	DDR clock mode recovery mode selection.
INIT	Binary	0, 1	1	Q initialization value.
SRTYPE	String	"SYNC", "ASYNC"	"SYNC"	Set/Reset type selection.

VHDL Instantiation Template

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

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


-- ODDR: Output Double Data Rate Output Register with Set, Reset
--       and Clock Enable.
--       Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

ODDR_inst : ODDR
generic map(
   DDR_CLK_EDGE => "OPPOSITE_EDGE", -- "OPPOSITE_EDGE" or "SAME_EDGE"
   INIT => '0',   -- Initial value for Q port ('1' or '0')
   SRTYPE => "SYNC") -- Reset Type ("ASYNC" or "SYNC")
port map (
   Q => Q,    -- 1-bit DDR output
   C => C,     -- 1-bit clock input
   CE => CE,   -- 1-bit clock enable input
   D1 => D1,   -- 1-bit data input (positive edge)
   D2 => D2,   -- 1-bit data input (negative edge)
   R => R,     -- 1-bit reset input
   S => S      -- 1-bit set input
);

-- End of ODDR_inst instantiation
```

Verilog Instantiation Template

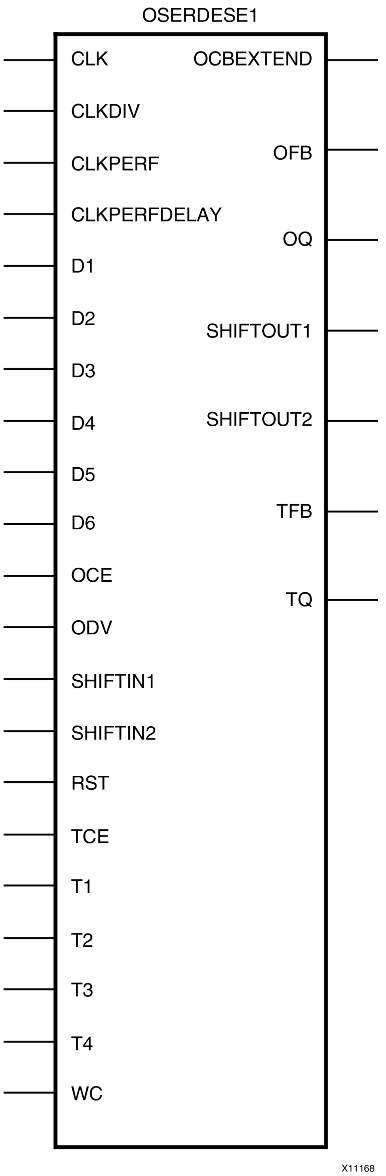
```
// ODDR: Output Double Data Rate Output Register with Set, Reset
//       and Clock Enable.
//       Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

ODDR #(
   .DDR_CLK_EDGE("OPPOSITE_EDGE"), // "OPPOSITE_EDGE" or "SAME_EDGE"
   .INIT(1'b0),    // Initial value of Q: 1'b0 or 1'b1
   .SRTYPE("SYNC") // Set/Reset type: "SYNC" or "ASYNC"
) ODDR_inst (
   .Q(Q),   // 1-bit DDR output
   .C(C),   // 1-bit clock input
   .CE(CE), // 1-bit clock enable input
   .D1(D1), // 1-bit data input (positive edge)
   .D2(D2), // 1-bit data input (negative edge)
   .R(R),   // 1-bit reset
   .S(S)    // 1-bit set
);

// End of ODDR_inst instantiation
```

For More Information

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

OSERDESE1

Primitive: Dedicated IOB Output Serializer



Introduction

This design element is a dedicated parallel-to-serial converter with specific clocking and logic resources designed to facilitate the implementation of high-speed source-synchronous interfaces. Every OSERDES module includes a dedicated serializer for data and 3-state control. Both data and 3-state serializers can be configured in SDR and DDR mode. Data serialization can be up to 6:1 (10:1 if using OSERDES Width Expansion). 3-state serialization can be up to 4:1. There is a dedicated DDR3 mode to support high-speed memory applications.

Port Descriptions

Port	Direction	Width	Function
CLK	Input	1	High Speed Clock Input - This clock input is used to drive the parallel-to-serial converters.
CLKDIV	Input	1	This divided high-speed clock input drives the parallel side of the parallel-to-serial converters. This clock is the divided version of the clock connected to the CLK port.
CLKPERF	Input	1	This port is part of a dedicated path that provides a high-performance clock from the MMCM to the OSERDESE1. CLKPERF is only available in MEMORY_DDR3 mode for DDR3 applications.
CLKPERFDELAY	Input	1	This port (CLKPERFDELAY) is part of a dedicated path that provides a high-performance clock from the MMCM delayed with the IODELAYE1 to OSERDESE1. CLKPERFDELAY is only available in MEMORY_DDR3 mode for DDR3 applications. When the IODELAYE1 is not being used to delay the CLKPERF, connect CLKPERFDELAY to the same source as CLKPERF.
D1 - D6	Input	1	Parallel Data Inputs - All incoming parallel data enters the OSERDES module through ports D1 to D6. These ports are connected to the FPGA fabric, and can be configured from two to six bits (i.e., a 6:1 serialization). Bit widths greater than six (up to 10) can be supported by using a second OSERDES in SLAVE mode.
OCBEXTEND	Output	1	Used in DDR3 mode to signal that the output circular buffer has extended the latency to match the CLK to the CLKPERF or CLKPERFDELAY.
OCE	Input	1	OCE is an active High clock enable for the data path.
ODV	Input	1	The ODV port is a part of the dedicated logic for the MEMORY_DDR3 mode. The ODV is asserted High by the user when CLKPERFDELAY delay through the IODELAYE1 is greater than half of the period. ODV is only available in MEMORY_DDR3 mode for DDR3 applications. When not using MEMORY_DDR3 mode, connect this port to GND.
OFB	Output	1	The output feedback port (OFB) is the serial (high-speed) data output port of the OSERDESE1 or the bypassed version of the CLKPERF. When the attribute ODELAYUSED is set to 0, the OFB port can be used to send out serial data to the ISERDESE1. When the attribute ODELAYUSED is set to 1 and the OSERDESE1 is in MEMORY_DDR3 mode, the OFB port can be used to link the high-performance clock input (CLKPERF) to the IODELAYE1.
OQ	Output	1	The OQ port is the data output port of the OSERDES module. Data at the input port D1 will appear first at OQ. This port connects the output of the data parallel-to-serial converter to the data input of the IOB. This port can not drive the IODELAYE1; the OFB pin must be used.

Port	Direction	Width	Function
RST	Input	1	The reset input causes the outputs of all data flip-flops in the CLK and CLKDIV domains to be driven Low asynchronously. OSERDES circuits running in the CLK domain where timing is critical use an internal, dedicated circuit to retime the RST input to produce a reset signal synchronous to the CLK domain. Similarly, there is a dedicated circuit to retime the RST input to produce a reset signal synchronous to the CLKDIV domain. Because there are OSERDES circuits that retime the RST input, the user is only required to provide a reset pulse to the RST input that meets timing on the CLKDIV frequency domain (synchronous to CLKDIV). Therefore, RST should be driven High for a minimum of one CLKDIV cycle. When building an interface consisting of multiple OSERDES ports, all OSERDES ports must be synchronized. The internal retiming of the RST input is designed so that all OSERDES blocks that receive the same reset pulse come out of reset synchronized with one another.
SHIFTIN1/ SHIFTIN2	Input	1	Cascade Input for data input expansion. Connect to SHIFTOUT1/2 of slave.
SHIFTOUT1/ SHIFTOUT2	Output	1	Cascade out for data input expansion. Connect to SHIFTIN1/2 of master.
TCE	Input	1	TCE is an active High clock enable for the 3-state control path.
TFB	Output	1	This port is the 3-state control output of the OSERDES module sent to the IODELAY. When used, this port connects the output of the 3-state parallel-to-serial converter to the control/3-state input of the IODELAY.
TQ	Output	1	This port is the 3-state control output of the OSERDES module. When used, this port connects the output of the 3-state parallel-to-serial converter to the control/3-state input of the IOB.
T1 - T4	Input	1	Parallel 3-State Inputs - All parallel 3-state signals enter the OSERDES module through ports T1 to T4. The ports are connected to the FPGA fabric, and can be configured as one, two, or four bits.
WC	Input	1	The WC port is a part of the dedicated logic for the MEMORY_DDR3 mode. The write command is issued when switching from writing to reading data. WC is only available in MEMORY_DDR3 mode for DDR3 applications. When not using MEMORY_DDR3 mode, connect this port to GND.

Design Entry Method

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

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
DATA_RATE_OQ	String	"DDR", "SDR"	"DDR"	Defines whether data (OQ) changes at every clock edge or every positive clock edge with respect to CLK.
DATA_RATE_TQ	String	"DDR", "BUF", "SDR"	"DDR"	Defines whether the 3-state (TQ) changes at every clock edge, every positive clock edge with respect to clock, or is set to buffer configuration.
DATA_WIDTH	Integer	4, 2, 3, 5, 6, 7, 8, 10	4	Defines the parallel-to-serial data converter width. This value also depends on the DATA_RATE_OQ value.If DATA_RATE_OQ = DDR, value is limited to 4, 6, 8, or 10. If DATA_RATE_OQ = SDR, value is limited to 2, 3, 4, 5, 6, 7, or 8.
DDR3_DATA	Integer	1, 0	1	For DDR3, if the I/O is a DQ or DQS pin, set to 1. If control, address, clock, etc. set to 0.
INIT_OQ	Binary	1'b0 to 1'b1	1'b0	Defines the initial value of OQ output.
INIT_TQ	Binary	1'b0 to 1'b1	1'b0	Defines the initial value of TQ output.
INTERFACE_TYPE	String	"DEFAULT", "MEMORY_DDR3"	"DEFAULT"	Chooses OSERDESE1 use model.
ODELAY_USED	Integer	0, 1	0	ODELAY_USED attribute is only for DDR3 mode. This attribute helps to set the output circular buffer in the correct mode when using ODELAY. Set ODELAY_USED to 0 for all other modes, even when ODELAY is used in the design.
SERDES_MODE	String	"MASTER", "SLAVE"	"MASTER"	Defines whether the OSERDES module is a master or slave when using width expansion.
SRVAL_OQ	Binary	1'b0 to 1'b1	1'b0	Defines the value of OQ outputs when the SR is invoked.
SRVAL_TQ	Binary	1'b0 to 1'b1	1'b0	Defines the value of TQ outputs when the SR is invoked.
TRISTATE_WIDTH	Integer	4, 1	4	Defines the parallel to serial 3-state converter width. If DATA_RATE_TQ = DDR, DATA_WIDTH = 4, and DATA_RATE_OQ = DDR, value is limited to 1 or 4. For all other settings of DATA_RATE_TQ, DATA_WIDTH, and DATA_RATE_OQ, value is limited to 1.

VHDL Instantiation Template

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

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


-- OSERDESE1: Output SERial/DESerializer
--            Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

OSERDESE1_inst : OSERDESE1
generic map (
   DATA_RATE_OQ => "DDR",         -- "SDR" or "DDR"
   DATA_RATE_TQ => "DDR",         -- "BUF", "SDR" or "DDR"
   DATA_WIDTH => 4,               -- Parallel data width (1-8,10)
   DDR3_DATA => 1,                -- Must leave at 1 (MIG-only parameter)
```

```
   INIT_OQ => '0',              -- Initial value of OQ output (0/1)
   INIT_TQ => '0',              -- Initial value of TQ output (0/1)
   INTERFACE_TYPE => "DEFAULT", -- Must leave at "DEFAULT" (MIG-only parameter)
   ODELAY_USED => 0,            -- Must leave at 0 (MIG-only parameter)
   SERDES_MODE => "MASTER",     -- "MASTER" or "SLAVE"
   SRVAL_OQ => '0',             -- OQ output value when SR is used (0/1)
   SRVAL_TQ => '0',             -- TQ output value when SR is used (0/1)
   TRISTATE_WIDTH => 4          -- Parallel to serial 3-state converter width (1 or 4)
)
port map (
   -- MIG-only Signals: 1-bit (each) output: Do not use unless generated by MIG
   OCBEXTEND => OCBEXTEND,       -- 1-bit output: Leave unconnected (MIG-only connected signal)
   -- Outputs: 1-bit (each) output: Serial output ports
   OFB => OFB,                   -- 1-bit output: Data feedback output to ISERDESE1
   OQ => OQ,                     -- 1-bit output: Data output (connect to I/O port)
   TFB => TFB,                   -- 1-bit output: 3-state control output
   TQ => TQ,                     -- 1-bit output: 3-state path output
   -- SHIFTOUT1-SHIFTOUT2: 1-bit (each) output: Data width expansion output ports
   SHIFTOUT1 => SHIFTOUT1,       -- 1-bit output: Connect to SHIFTIN1 of slave or unconnected
   SHIFTOUT2 => SHIFTOUT2,       -- 1-bit output: Connect to SHIFTIN2 of slave or unconnected
   -- Clocks: 1-bit (each) input: OSERDESE1 clock input ports
   CLK => CLK,                   -- 1-bit input: High-speed clock input
   CLKDIV => CLKDIV,             -- 1-bit input: Divided clock input
   -- Control Signals: 1-bit (each) input: Clock enable and reset input ports
   OCE => OCE,                   -- 1-bit input: Active high clock data path enable input
   RST => RST,                   -- 1-bit input: Active high reset input
   TCE => TCE,                   -- 1-bit input: Active high clock enable input for 3-state
   -- D1 - D6: 1-bit (each) input: Parallel data inputs
   D1 => D1,
   D2 => D2,
   D3 => D3,
   D4 => D4,
   D5 => D5,
   D6 => D6,
   -- MIG-only Signals: 1-bit (each) input: Do not use unless generated by MIG
   CLKPERF => CLKPERF,           -- 1-bit input: Ground input (MIG-only connected signal)
   CLKPERFDELAY => CLKPERFDELAY, -- 1-bit input: Ground input (MIG-only connected signal)
   ODV => ODV,                   -- 1-bit input: Ground input (MIG-only connected signal)
   WC => WC,                     -- 1-bit input: Ground input (MIG-only connected signal)
   -- SHIFTIN1-SHIFTIN2: 1-bit (each) input: Data width expansion input ports
   SHIFTIN1 => SHIFTIN1,         -- 1-bit input: Connect to SHIFTOUT1 of master or GND
   SHIFTIN2 => SHIFTIN2,         -- 1-bit input: Connect to SHIFTOUT2 of master or GND
   -- T1 - T4: 1-bit (each) input: Parallel 3-state inputs
   T1 => T1,
   T2 => T2,
   T3 => T3,
   T4 => T4
);

-- End of OSERDESE1_inst instantiation
```

Verilog Instantiation Template

```
// OSERDESE1: Output SERial/DESerializer
//            Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

OSERDESE1 #(
   .DATA_RATE_OQ("DDR"),       // "SDR" or "DDR"
   .DATA_RATE_TQ("DDR"),       // "BUF", "SDR" or "DDR"
   .DATA_WIDTH(4),             // Parallel data width (1-8,10)
   .DDR3_DATA(1),              // Must leave at 1 (MIG-only parameter)
   .INIT_OQ(1'b0),             // Initial value of OQ output (0/1)
   .INIT_TQ(1'b0),             // Initial value of TQ output (0/1)
   .INTERFACE_TYPE("DEFAULT"), // Must leave at "DEFAULT" (MIG-only parameter)
   .ODELAY_USED(0),            // Must leave at 0 (MIG-only parameter)
   .SERDES_MODE("MASTER"),     // "MASTER" or "SLAVE"
   .SRVAL_OQ(1'b0),            // OQ output value when SR is used (0/1)
   .SRVAL_TQ(1'b0),            // TQ output value when SR is used (0/1)
   .TRISTATE_WIDTH(4)          // Parallel to serial 3-state converter width (1 or 4)
)
OSERDESE1_inst (
   // MIG-only Signals: 1-bit (each) output: Do not use unless generated by MIG
   .OCBEXTEND(OCBEXTEND),       // 1-bit output: Leave unconnected (MIG-only connected signal)
   // Outputs: 1-bit (each) output: Serial output ports
   .OFB(OFB),                   // 1-bit output: Data feedback output to ISERDESE1
   .OQ(OQ),                     // 1-bit output: Data output (connect to I/O port)
   .TFB(TFB),                   // 1-bit output: 3-state control output
   .TQ(TQ),                     // 1-bit output: 3-state path output
   // SHIFTOUT1-SHIFTOUT2: 1-bit (each) output: Data width expansion output ports
   .SHIFTOUT1(SHIFTOUT1),       // 1-bit output: Connect to SHIFTIN1 of slave or unconnected
   .SHIFTOUT2(SHIFTOUT2),       // 1-bit output: Connect to SHIFTIN2 of slave or unconnected
   // Clocks: 1-bit (each) input: OSERDESE1 clock input ports
   .CLK(CLK),                   // 1-bit input: High-speed clock input
   .CLKDIV(CLKDIV),             // 1-bit input: Divided clock input
   // Control Signals: 1-bit (each) input: Clock enable and reset input ports
   .OCE(OCE),                   // 1-bit input: Active high clock data path enable input
   .RST(RST),                   // 1-bit input: Active high reset input
   .TCE(TCE),                   // 1-bit input: Active high clock enable input for 3-state
   // D1 - D6: 1-bit (each) input: Parallel data inputs
   .D1(D1),
   .D2(D2),
   .D3(D3),
   .D4(D4),
   .D5(D5),
   .D6(D6),
   // MIG-only Signals: 1-bit (each) input: Do not use unless generated by MIG
   .CLKPERF(CLKPERF),           // 1-bit input: Ground input (MIG-only connected signal)
   .CLKPERFDELAY(CLKPERFDELAY), // 1-bit input: Ground input (MIG-only connected signal)
   .ODV(ODV),                   // 1-bit input: Ground input (MIG-only connected signal)
   .WC(WC),                     // 1-bit input: Ground input (MIG-only connected signal)
   // SHIFTIN1-SHIFTIN2: 1-bit (each) input: Data width expansion input ports
   .SHIFTIN1(SHIFTIN1),         // 1-bit input: Connect to SHIFTOUT1 of master or GND
   .SHIFTIN2(SHIFTIN2),         // 1-bit input: Connect to SHIFTOUT2 of master or GND
   // T1 - T4: 1-bit (each) input: Parallel 3-state inputs
   .T1(T1),
   .T2(T2),
   .T3(T3),
   .T4(T4)
);

// End of OSERDESE1_inst instantiation
```

For More Information

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

PCIE_2_0

Primitive: PCI Express version 2.0 compliant port

Introduction

This design element is intended for use in conjunction with other resources located in the FPGA, such as the RocketIO™ transceivers, block RAMs, and various clocking resources. To implement an Endpoint, Root Port, or custom PCI EXPRESS® design using PCIe_2_0, 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 instantiates the PCIE_2_0 software primitive, connects the interfaces to the correct FPGA resources, sets all attributes, and presents a simple, user-friendly interface.

Design Entry Method

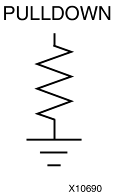
To instantiate this component, use the PCI EXPRESS core or an associated core containing the component. Xilinx does not recommend direct instantiation of this component.

For More Information

- See the *Virtex-6 FPGA RocketIO GTX Transceivers User Guide (UG366)*.
- See the *Virtex-6 FPGA User Documentation (User Guides and Data Sheets)*.

PULLDOWN

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



Introduction

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

Port Descriptions

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

Design Entry Method

Instantiation	Yes
Inference	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
--           Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

-- End of PULLDOWN_inst instantiation
```

Verilog Instantiation Template

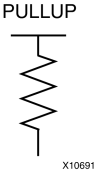
```
// PULLDOWN: I/O Buffer Weak Pull-down
//           Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

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

// End of PULLDOWN_inst instantiation
```

For More Information

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

PULLUP

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



Introduction

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

Port Descriptions

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

Design Entry Method

Instantiation	Yes
Inference	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
--         Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

-- End of PULLUP_inst instantiation
```

Verilog Instantiation Template

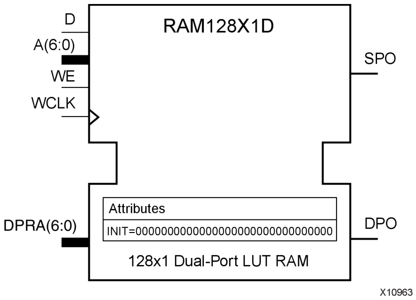
```
// PULLUP: I/O Buffer Weak Pull-up
//         Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

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

// End of PULLUP_inst instantiation
```

For More Information

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

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
--            Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

RAM128X1D_inst : RAM128X1D
generic map (
   INIT => X"00000000000000000000000000000000")
port map (
   DPO => DPO,     -- Read/Write port 1-bit output
   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

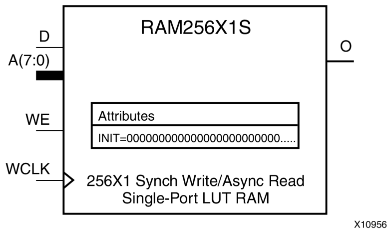
```
// RAM128X1D: 128-deep by 1-wide positive edge write, asynchronous read
//            dual-port distributed LUT RAM
//            Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

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

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
--            Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

RAM256X1S_inst : RAM256X1S
generic map (
   INIT => X"0000000000000000000000000000000000000000000000000000000000000000")
port map (
   O => O,  -- Read/Write port 1-bit output
   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

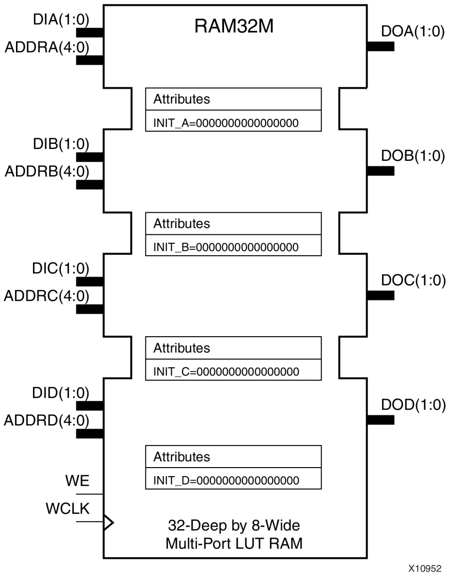
```
// RAM256X1S: 256-deep by 1-wide positive edge write, asynchronous read
//            single-port distributed LUT RAM
//            Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

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

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
--         Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

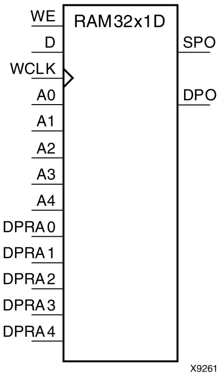
```
// RAM32M: 32-deep by 8-wide Multi Port LUT RAM
//         Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

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

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
--             Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

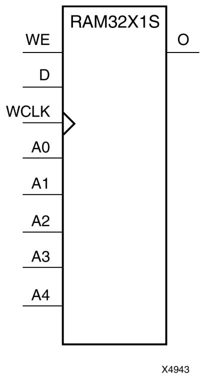
```
// RAM32X1D: 32 x 1 positive edge write, asynchronous read dual-port distributed RAM
//           Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

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 *Virtex-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
--           Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

RAM32X1S_inst : RAM32X1S
generic map (
   INIT => X"00000000")
port map (
   O => O,       -- RAM output
   A0 => A0,     -- RAM address[0] input
   A1 => A1,     -- RAM address[1] input
   A2 => A2,     -- RAM address[2] input
   A3 => A3,     -- RAM address[3] input
   A4 => A4,     -- RAM address[4] input
   D => D,       -- RAM data input
   WCLK => WCLK, -- Write clock input
   WE => WE      -- Write enable input
);

-- End of RAM32X1S_inst instantiation
```

Verilog Instantiation Template

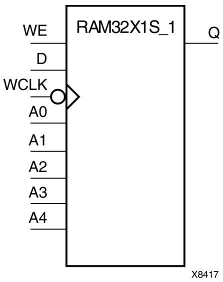
```
// RAM32X1S: 32 x 1 posedge write distributed (LUT) RAM
//           Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

RAM32X1S #(
   .INIT(32'h00000000)  // Initial contents of RAM
) RAM32X1S_inst (
   .O(O),       // RAM output
   .A0(A0),     // RAM address[0] input
   .A1(A1),     // RAM address[1] input
   .A2(A2),     // RAM address[2] input
   .A3(A3),     // RAM address[3] input
   .A4(A4),     // RAM address[4] input
   .D(D),       // RAM data input
   .WCLK(WCLK), // Write clock input
   .WE(WE)      // Write enable input
);

// End of RAM32X1S_inst instantiation
```

For More Information

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

RAM32X1S_1

Primitive: 32-Deep by 1-Wide Static Synchronous RAM with Negative-Edge Clock



Introduction

The design element is a 32-word by 1-bit static random access memory with synchronous write capability. When the write enable is Low, transitions on the write clock (WCLK) are ignored and data stored in the RAM is not affected. When (WE) is High, any negative transition on (WCLK) loads the data on the data input (D) into the word selected by the 5-bit address (A4:A0). For predictable performance, address and data inputs must be stable before a High-to-Low (WCLK) transition. This RAM block assumes an active-Low (WCLK). However, (WCLK) can be active-High or active-Low. Any inverter placed on the (WCLK) input net is absorbed into the block.

The signal output on the data output pin (O) is the data that is stored in the RAM at the location defined by the values on the address pins. You can initialize RAM32X1S_1 during configuration using the INIT attribute.

Logic Table

Inputs			Outputs
WE (Mode)	**WCLK**	**D**	**O**
0 (read)	X	X	Data
1 (read)	0	X	Data
1 (read)	1	X	Data
1 (write)	↓	D	D
1 (read)	↑	X	Data
Data = word addressed by bits A4:A0			

Design Entry Method

Instantiation	Yes
Inference	Recommended
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
--             Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

RAM32X1S_1_inst : RAM32X1S_1
generic map (
   INIT => X"00000000")
port map (
   O => O,       -- RAM output
   A0 => A0,     -- RAM address[0] input
   A1 => A1,     -- RAM address[1] input
   A2 => A2,     -- RAM address[2] input
   A3 => A3,     -- RAM address[3] input
   A4 => A4,     -- RAM address[4] input
   D => D,       -- RAM data input
   WCLK => WCLK, -- Write clock input
   WE => WE      -- Write enable input
);

-- End of RAM32X1S_1_inst instantiation
```

Verilog Instantiation Template

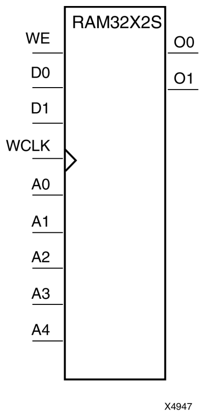
```
// RAM32X1S_1: 32 x 1 negedge write distributed (LUT) RAM
//             Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

RAM32X1S_1 #(
   .INIT(32'h00000000)  // Initial contents of RAM
)RAM32X1S_1_inst (
   .O(O),       // RAM output
   .A0(A0),     // RAM address[0] input
   .A1(A1),     // RAM address[1] input
   .A2(A2),     // RAM address[2] input
   .A3(A3),     // RAM address[3] input
   .A4(A4),     // RAM address[4] input
   .D(D),       // RAM data input
   .WCLK(WCLK), // Write clock input
   .WE(WE)      // Write enable input
);

// End of RAM32X1S_1_inst instantiation
```

For More Information

See the *Virtex-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
--           Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

RAM32X2S_inst : RAM32X2S
generic map (
   INIT_00 => X"00000000", --  INIT for bit 0 of RAM
   INIT_01 => X"00000000") --  INIT for bit 1 of RAM
port map (
   O0 => O0,      -- RAM data[0] output
   O1 => O1,      -- RAM data[1] output
   A0 => A0,      -- RAM address[0] input
   A1 => A1,      -- RAM address[1] input
   A2 => A2,      -- RAM address[2] input
   A3 => A3,      -- RAM address[3] input
   A4 => A4,      -- RAM address[4] input
   D0 => D0,      -- RAM data[0] input
   D1 => D1,      -- RAM data[1] input
   WCLK => WCLK, -- Write clock input
   WE => WE       -- Write enable input
);

-- End of RAM32X2S_inst instantiation
```

Verilog Instantiation Template

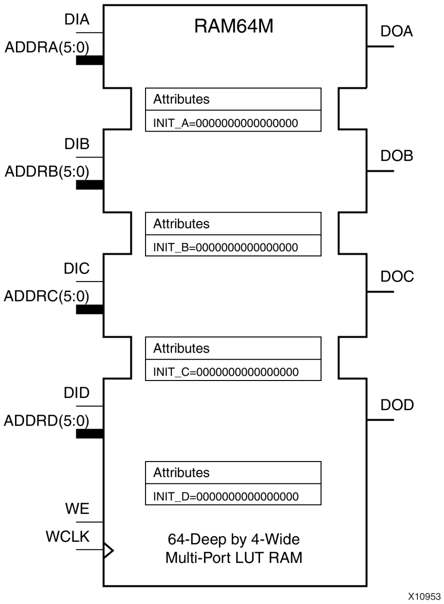
```
// RAM32X2S: 32 x 2 posedge write distributed (LUT) RAM
//           Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

RAM32X2S #(
   .INIT_00(32'h00000000), // INIT forr bit 0 of RAM
   .INIT_01(32'h00000000)  // INIT forr bit 1 of RAM
) RAM32X2S_inst (
   .O0(O0),      // RAM data[0] output
   .O1(O1),      // RAM data[1] output
   .A0(A0),      // RAM address[0] input
   .A1(A1),      // RAM address[1] input
   .A2(A2),      // RAM address[2] input
   .A3(A3),      // RAM address[3] input
   .A4(A4),      // RAM address[4] input
   .D0(D0),      // RAM data[0] input
   .D1(D1),      // RAM data[1] input
   .WCLK(WCLK), // Write clock input
   .WE(WE)       // Write enable input
);

// End of RAM32X2S_inst instantiation
```

For More Information

See the *Virtex-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
--          Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

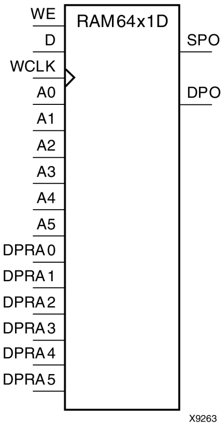
```
// RAM64M: 64-deep by 4-wide Multi Port LUT RAM
//         Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

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

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
--           Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

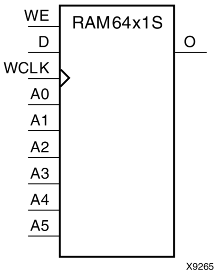
```
// RAM64X1D: 64 x 1 positive edge write, asynchronous read dual-port distributed RAM
//           Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

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

RAM64X1S

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



Introduction

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

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

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

Logic Table

Mode selection is shown in the following logic table

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

Design Entry Method

Instantiation	Yes
Inference	Recommended
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
--           Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

-- End of RAM64X1S_inst instantiation
```

Verilog Instantiation Template

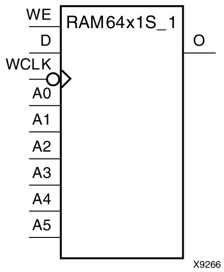
```
// RAM64X1S: 64 x 1 positive edge write, asynchronous read single-port distributed RAM
//           Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

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

// End of RAM64X1S_inst instantiation
```

For More Information

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

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
--             Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

RAM64X1S_1_inst : RAM64X1S_1
generic map (
   INIT => X"0000000000000000")
port map (
   O => O,         -- 1-bit data output
   A0 => A0,       -- Address[0] input bit
   A1 => A1,       -- Address[1] input bit
   A2 => A2,       -- Address[2] input bit
   A3 => A3,       -- Address[3] input bit
   A4 => A4,       -- Address[4] input bit
   A5 => A5,       -- Address[5] input bit
   D => D,         -- 1-bit data input
   WCLK => WCLK,   -- Write clock input
   WE => WE        -- Write enable input
);

-- End of RAM64X1S_1_inst instantiation
```

Verilog Instantiation Template

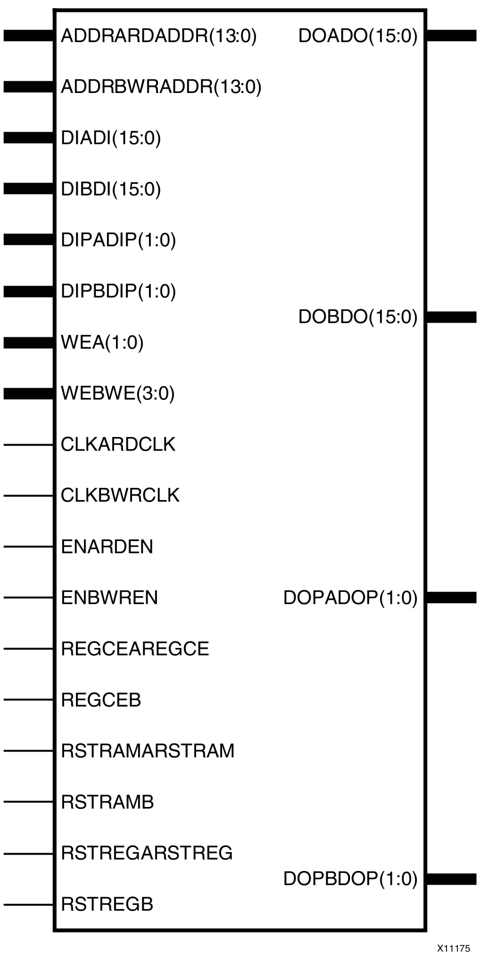
```
// RAM64X1S_1: 64 x 1 negative edge write, asynchronous read single-port distributed RAM
//             Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

RAM64X1S_1 #(
   .INIT(64'h0000000000000000) // Initial contents of RAM
) RAM64X1S_1_inst (
   .O(O),        // 1-bit data output
   .A0(A0),      // Address[0] input bit
   .A1(A1),      // Address[1] input bit
   .A2(A2),      // Address[2] input bit
   .A3(A3),      // Address[3] input bit
   .A4(A4),      // Address[4] input bit
   .A5(A5),      // Address[5] input bit
   .D(D),        // 1-bit data input
   .WCLK(WCLK),  // Write clock input
   .WE(WE)       // Write enable input
);

// End of RAM64X1S_1_inst instantiation
```

For More Information

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

RAMB18E1

Primitive: 18K-bit Configurable Synchronous Block RAM



Introduction

Virtex®-6 devices contain several block RAM memories that can be configured as FIFOs, automatic error correction RAM, or general-purpose 36 kb or 18 kb RAM/ROM memories. These block RAM memories offer fast and flexible storage of large amounts of on-chip data. This element allows access to the block RAM in the 18 kb configuration. This element can be configured and used as a 1-bit wide by 16K deep to an 18-bit wide by 1029-bit deep true dual port RAM. This element can also be configured as a 36-bit wide by 512 deep simple dual port RAM. Both read and write operations are fully synchronous to the supplied clock(s) in the component. However, the READ and WRITE ports can operate fully independently and asynchronously to each other, accessing the same memory array. When configured in the wider data width modes, 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 Descriptions

Port	Direction	Width	Function
ADDRARDADDR[13:0]	Input	14	Port A address input bus/Read address input bus.
ADDRBWRADDR[13:0]	Input	14	Port B address input bus/Write address input bus.
CLKARDCLK	Input	1	Port A clock input/Read clock input.

Port	Direction	Width	Function
CLKBWRCLK	Input	1	Port B clock input/Write clock input.
DIADI[15:0]	Input	16	Port A data input bus/Data input bus addressed by WRADDR. When RAM_MODE=SDP, DIADI is the logical DI[15:0].
DIBDI[15:0]	Input	16	Port B data input bus/Data input bus addressed by WRADDR. When RAM_MODE=SDP, DIBDI is the logical DI[31:16].
DIPADIP[1:0]	Input	2	Port A parity data input bus/Data parity input bus addressed by WRADDR. When RAM_MODE=SDP, DIPADIP is the logical DIP[1:0].
DIPBDIP[1:0]	Input	2	Port B parity data input bus/Data parity input bus addressed by WRADDR. When RAM_MODE=SDP, DIPBDIP is the logical DIP[3:2].
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].
ENARDEN	Input	1	Port A RAM enable/Read enable.
ENBWREN	Input	1	Port B RAM enable/Write enable.
REGCEAREGCE	Input	1	Port A output register clock enable input/Output register clock enable input (valid only when DO_REG=1).
REGCEB	Input	1	Port B output register clock enable (valid only when DO_REG=1 and RAM_MODE=TDP).
RSTRAMARSTRAM	Input	1	Synchronous data latch set/reset to value indicated by SRVAL_A. RSTRAMARSTRAM sets/resets the BRAM data output latch when DO_REG=0 or 1. If DO_REG=1 there is a cycle of latency between the internal data latch node that is reset by RSTRAMARSTRAM and the DO output of the BRAM. This signal is RSTRAMA on port A when RAM_MODE=TDP and RSTRAM when RAM_MODE=SDP.
RSTRAMB	Input	1	Synchronous data latch set/reset to value indicated by SRVAL_B. RSTRAMB sets/resets the BRAM data output latch when DO_REG=0 or 1. If DO_REG=1 there is a cycle of latency between the internal data latch node that is reset by RSTRAMB and the DO output of the BRAM. Not used when RAM_MODE=SDP.
RSTREGARSTREG	Input	1	Synchronous output register set/reset to value indicated by SRVAL_A. RSTREGARSTREG sets/resets the output register when DO_REG=1. RSTREG_PRIORITY_A determines if this signal gets priority over REGCEAREGCE. This signal is RSTREGA on port A when RAM_MODE=TDP and RSTREG when RAM_MODE=SDP.
RSTREGB	Input	1	Synchronous output register set/reset to value indicated by SRVAL_B. RSTREGB sets/resets the output register when DO_REG=1. RSTREG_PRIORITY_B determines if this signal gets priority over REGCEB. Not used when RAM_MODE=SDP.

Port	Direction	Width	Function
WEA[1:0]	Input	2	Port A byte-wide write enable. Not used when RAM_MODE=SDP. See User Guide for WEA mapping for different port widths.
WEBWE[3:0]	Input	4	Port B byte-wide write enable/Write enable. See User Guide for WEBWE mapping for different port widths.

Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	Yes
Macro support	Yes

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
COLLISION CHECK	String	"ALL", "GENERATE_X_ ONLY", "NONE", "WARNING_ONLY"	"ALL"	Allows modification of the simulation behavior so that if a memory collision occurs: • "ALL" - warning produced and affected outputs/memory location go unknown (X) • "WARNING_ONLY" - warning produced and affected outputs/memory retain last value • "GENERATE_X_ONLY" - no warning, however affected outputs/memory go unknown (X) • "NONE" - no warning and affected outputs/memory retain last value **Note** Setting this to a value other than ALL can allow problems in the design to go unnoticed during simulation. Care should be taken when changing the value of this attribute.
DOA_REG	Integer	0, 1	0	A value of 1 enables 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 result in slower clock-to-out timing. Applies to port A in TDP mode and up to 18 lower bits (including parity bits) in SDP mode.

Attribute	Data Type	Allowed Values	Default	Description
DOB_REG	Integer	0, 1	0	A value of 1 enables 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 result in slower clock-to-out timing. Applies to port B in TDP mode and upper bits (including parity bits) in SDP mode.
INIT_A	Hexa-decimal	Any 18 Bit Value	All zeros	Specifies the initial value on the Port A output after configuration. Applies to port A in TDP mode and up to 18 lower bits (including parity bits) in SDP mode.
INIT_B	Hexa-decimal	Any 18 Bit Value	All zeros	Specifies the initial value on the Port B output after configuration. Applies to port B in TDP mode and upper bits (including parity bits) in SDP mode.
INIT_FILE	String	String representing file name and location	None	File name of file used to specify initial RAM contents.
INIT_00 - INIT_3F	Hexa-decimal	All zeros to all ones	All zeros	Allows specification of the initial contents of the 16 kb data memory array.
INITP_00 - INITP_07	Hexa-decimal	All zeros to all ones	All zeros	Allows specification of the initial contents of the 2 kb parity data memory array.
RAM_MODE	String	"TDP", "SDP"	"TDP"	Selects simple dual port (SDP) or true dual port (TDP) mode.
RDADDR_ COLLISION _HWCONFIG	String	"DELAYED_WRITE", "PERFORMANCE"	"DELAYED_WRITE"	• Setting to "PERFORMANCE" allows for higher clock performance (frequency) in READ_FIRST mode. • If using the same clock on both ports of the RAM with "PERFORMANCE" mode, the address overlap collision rules apply. • In "DELAYED_WRITE" mode, you can safely use the BRAM without incurring collisions. • Not supported for ES silicon and must be set to "DELAYED_WRITE" if targeting ES devices.
READ_WIDTH_A	Integer	0, 1, 2, 4, 9, 18, 36, 72	0	Specifies the desired data width for a read on Port A, including parity bits. This value must be 0 if the Port A is not used. Otherwise, it should be set to the desired port width. In SDP mode, this is the read width including parity bits.

Attribute	Data Type	Allowed Values	Default	Description
READ_WIDTH_B	Integer	0, 1, 2, 4, 9, 18	0	Specifies the desired data width for a read on Port B including parity bits. This value must be 0 if the Port B is not used. Otherwise, it should be set to the desired port width. Not used for SDP mode.
RSTREG_PRIORITY_A	String	"RSTREG", "REGCE"	"RSTREG"	Selects register priority for RSTREG or REGCE. Applies to port A in TDP mode and up to 18 lower bits (including parity bits) in SDP mode.
RSTREG_PRIORITY_B	String	"RSTREG", "REGCE"	"RSTREG"	Selects register priority for RSTREG or REGCE. Applies to port B in TDP mode and upper bits (including parity bits) in SDP mode.
SRVAL_A	Hexa-decimal	Any 18 Bit Value	All zeros	Specifies the output value of the RAM upon assertion of the synchronous reset (RSTREG) signal. Applies to port A in TDP mode and up to 18 lower bits (including parity bits) in SDP mode.
SRVAL_B	Hexa-decimal	Any 18 Bit Value	All zeros	Specifies the output value of the RAM upon assertion of the synchronous reset (RSTREG) signal. Applies to port B in TDP mode and upper bits (including parity bits) in SDP mode.
WRITEMODE	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_WIDTH_A	Integer	0, 1, 2, 4, 9, 18	0	Specifies the desired data width for a write to Port A including parity bits. This value must be 0 if the port is not used. Otherwise should be set to the desired write width. Not used in SDP mode.
WRITE_WIDTH_B	Integer	0, 1, 2, 4, 9, 18, 36, 72	0	Specifies the desired data width for a write to Port B including parity bits. This value must be 0 if the port is not used. Otherwise should be set to the desired write width. In SDP mode, this is the write width including parity bits.

VHDL Instantiation Template

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

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

```
-- RAMB18E1: 18K-bit Configurable Synchronous Block RAM
--           Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

RAMB18E1_inst : RAMB18E1
generic map (
   -- Colision check: Values ("ALL", "WARNING_ONLY", "GENERATE_X_ONLY" or "NONE")
   SIM_COLLISION_CHECK => "ALL",
   -- DOA_REG, DOB_REG: Optional output register (0 or 1)
   DOA_REG => 0,
   DOB_REG => 0,
   -- INITP_00 to INITP_07: Initial contents of parity memory array
   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 contents of data memory array
   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 ports
   INIT_A => X"00000",
   INIT_B => X"00000",
   INIT_FILE => "NONE",                                                            -- RAM init file
   RAM_MODE => "TDP",                                                              -- "SDP" or "TDP"
   RDADDR_COLLISION_HWCONFIG => "DELAYED_WRITE",                                   -- "PERFORMANCE" or
                                                                                   -- "DELAYED_WRITE"
   -- READ_WIDTH_A/B, WRITE_WIDTH_A/B: Read/write width per port
   READ_WIDTH_A => 0,                                                              -- 0,1,2,4,9,18,36
   READ_WIDTH_B => 0,                                                              -- 0,1,2,4,9,18
   WRITE_WIDTH_A => 0,                                                             -- 0,1,2,4,9,18
   WRITE_WIDTH_B => 0,                                                             -- 0,1,2,4,9,18,36
   -- RSTREG_PRIORITY_A, RSTREG_PRIORITY_B: Reset or enable priority ("RSTREG" or "REGCE")
   RSTREG_PRIORITY_A => "RSTREG",
   RSTREG_PRIORITY_B => "RSTREG",
   -- SRVAL_A, SRVAL_B: Set/reset value for output
   SRVAL_A => X"00000",
   SRVAL_B => X"00000",
   -- WriteMode: Value on output upon a write ("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: 14-bit (each) input: Port A address and control signals (read port
   -- when RAM_MODE="SDP")
   ADDRARDADDR => ADDRARDADDR,     -- 14-bit input: A port address/Read address input
   CLKARDCLK => CLKARDCLK,         -- 1-bit input: A port clock/Read clock input
   ENARDEN => ENARDEN,             -- 1-bit input: A port enable/Read enable input
   REGCEAREGCE => REGCEAREGCE,     -- 1-bit input: A port register enable/Register enable input
   RSTRAMARSTRAM => RSTRAMARSTRAM, -- 1-bit input: A port set/reset input
   RSTREGARSTREG => RSTREGARSTREG, -- 1-bit input: A port register set/reset input
   WEA => WEA,                     -- 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: 14-bit (each) input: Port B address and control signals (write port
   -- when RAM_MODE="SDP")
   ADDRBWRADDR => ADDRBWRADDR,     -- 14-bit input: B port address/Write address input
   CLKBWRCLK => CLKBWRCLK,         -- 1-bit input: B port clock/Write clock input
   ENBWREN => ENBWREN,             -- 1-bit input: B port enable/Write enable input
   REGCEB => REGCEB,               -- 1-bit input: B port register enable input
   RSTRAMB => RSTRAMB,             -- 1-bit input: B port set/reset input
   RSTREGB => RSTREGB,             -- 1-bit input: B port register set/reset input
   WEBWE => WEBWE,                 -- 4-bit input: B port write enable/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 RAMB18E1_inst instantiation
```

Verilog Instantiation Template

```
// RAMB18E1: 18K-bit Configurable Synchronous Block RAM
//           Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

RAMB18E1 #(
   // Colision check: Values ("ALL", "WARNING_ONLY", "GENERATE_X_ONLY" or "NONE")
   .SIM_COLLISION_CHECK("ALL"),
   // DOA_REG, DOB_REG: Optional output register (0 or 1)
   .DOA_REG(0),
   .DOB_REG(0),
   // INITP_00 to INITP_07: Initial contents of parity memory array
   .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 contents of data memory array
   .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),
```

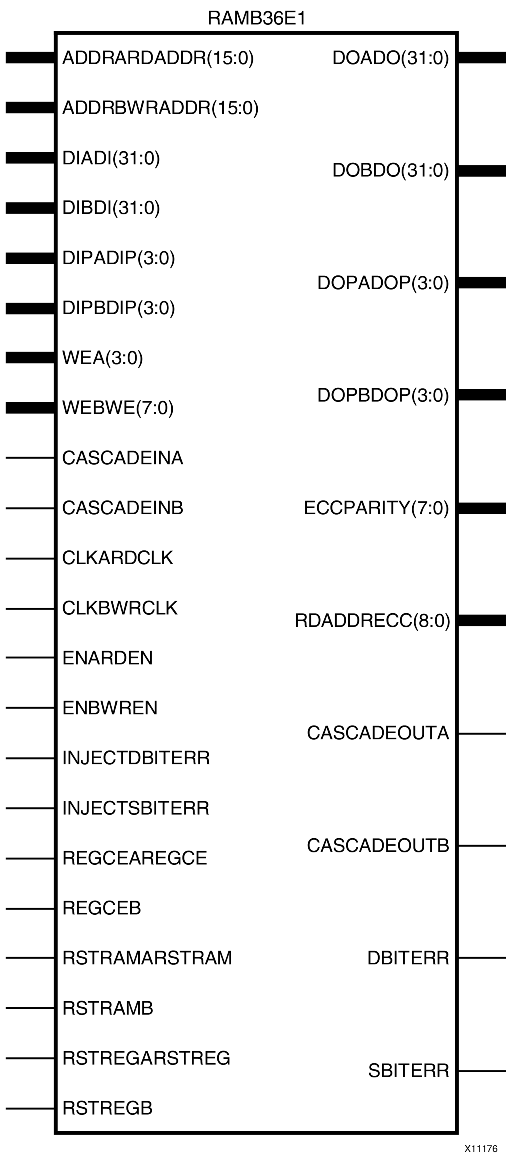
```
   .INIT_32(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_33(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_34(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INIT_35(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .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 ports
   .INIT_A(18'h00000),
   .INIT_B(18'h00000),
   .INIT_FILE("NONE"),                                                  // RAM init file
   .RAM_MODE("TDP"),                                                    // "SDP" or "TDP"
   .RDADDR_COLLISION_HWCONFIG("DELAYED_WRITE"),                         // "PERFORMANCE" or
                                                                        // "DELAYED_WRITE"
   // READ_WIDTH_A/B, WRITE_WIDTH_A/B: Read/write width per port
   .READ_WIDTH_A(0),                                                    // 0,1,2,4,9,18,36
   .READ_WIDTH_B(0),                                                    // 0,1,2,4,9,18
   .WRITE_WIDTH_A(0),                                                   // 0,1,2,4,9,18
   .WRITE_WIDTH_B(0),                                                   // 0,1,2,4,9,18,36
   // RSTREG_PRIORITY_A, RSTREG_PRIORITY_B: Reset or enable priority ("RSTREG" or "REGCE")
   .RSTREG_PRIORITY_A("RSTREG"),
   .RSTREG_PRIORITY_B("RSTREG"),
   // SRVAL_A, SRVAL_B: Set/reset value for output
   .SRVAL_A(18'h00000),
   .SRVAL_B(18'h00000),
   // WriteMode: Value on output upon a write ("WRITE_FIRST", "READ_FIRST", or "NO_CHANGE")
   .WRITE_MODE_A("WRITE_FIRST"),
   .WRITE_MODE_B("WRITE_FIRST")
)
RAMB18E1_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: 14-bit (each) input: Port A address and control signals (read port
   // when RAM_MODE="SDP")
   .ADDRARDADDR(ADDRARDADDR),     // 14-bit input: A port address/Read address input
   .CLKARDCLK(CLKARDCLK),         // 1-bit input: A port clock/Read clock input
   .ENARDEN(ENARDEN),             // 1-bit input: A port enable/Read enable input
   .REGCEAREGCE(REGCEAREGCE),     // 1-bit input: A port register enable/Register enable input
   .RSTRAMARSTRAM(RSTRAMARSTRAM), // 1-bit input: A port set/reset input
   .RSTREGARSTREG(RSTREGARSTREG), // 1-bit input: A port register set/reset input
   .WEA(WEA),                     // 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: 14-bit (each) input: Port B address and control signals (write port
   // when RAM_MODE="SDP")
   .ADDRBWRADDR(ADDRBWRADDR),     // 14-bit input: B port address/Write address input
   .CLKBWRCLK(CLKBWRCLK),         // 1-bit input: B port clock/Write clock input
   .ENBWREN(ENBWREN),             // 1-bit input: B port enable/Write enable input
   .REGCEB(REGCEB),               // 1-bit input: B port register enable input
   .RSTRAMB(RSTRAMB),             // 1-bit input: B port set/reset input
   .RSTREGB(RSTREGB),             // 1-bit input: B port register set/reset input
   .WEBWE(WEBWE),                 // 4-bit input: B port write enable/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 RAMB18E1_inst instantiation
```

For More Information

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

RAMB36E1

Primitive: 36K-bit Configurable Synchronous Block RAM



Introduction

Virtex®-6 devices contain several block RAM memories that can be configured as FIFOs, automatic error-correction RAM, or general-purpose 36 kb or 18 kb RAM/ROM memories. These Block RAM memories offer fast and flexible storage of large amounts of on-chip data. This element allows access to the Block RAM in the 36 kb configurations. This element can be cascaded to create a larger RAM. This component can be configured and used as a 1-bit wide by 32K deep to a 36-bit wide by 1K deep true dual port RAM. Both read and write operations are fully synchronous to the supplied clock(s) in the component. However, the READ and WRITE ports can operate fully independently and asynchronously of each other, accessing the same memory array. When configured in the wider data width modes, byte-enable write operations are possible, and an optional output register can be used to reduce the clock-to-out times of the RAM. Error detection and correction circuitry can also be enabled to uncover and rectify possible memory corruption.

Port Descriptions

Port	Direction	Width	Function
ADDRARDADDR[15:0]	Input	16	Port A address input bus/Read address input bus.
ADDRBWRADDR[15:0]	Input	16	Port B address input bus/Write address input bus.
CASCADEINA	Input	1	Port A cascade input. Never use when RAM_MODE=SDP.
CASCADEINB	Input	1	Port B cascade input. Never use when RAM_MODE=SDP.
CASCADEOUTA	Output	1	Port A cascade output. Never use when RAM_MODE=SDP.
CASCADEOUTB	Output	1	Port B cascade output. Never use when RAM_MODE=SDP.
CLKARDCLK	Input	1	Port A clock input/Read clock input.
CLKBWRCLK	Input	1	Port B clock input/Write clock input.
DBITERR	Output	1	Status output from ECC function to indicate a double bit error was detected. Set EN_ECC_READ to TRUE to use this functionality. Not used when RAM_MODE=TDP.
DIADI[31:0]	Input	32	Port A data input bus/Data input bus addressed by WRADDR. When RAM_MODE=SDP, DIADI is the logical DI[31:0].
DIBDI[31:0]	Input	32	Port B data input bus/Data input bus addressed by WRADDR. When RAM_MODE=SDP, DIBDI is the logical DI[63:32].
DIPADIP[3:0]	Input	4	Port A parity data input bus/Data parity input bus addressed by WRADDR. When RAM_MODE=SDP, DIPADIP is the logical DIP[3:0].
DIPBDIP[3:0]	Input	4	Port B parity data input bus/Data parity input bus addressed by WRADDR. When RAM_MODE=SDP, DIPBDIP is the logical DIP[7:4].
DOADO[31:0]	Output	32	Port A data output bus/Data output bus addressed by RDADDR. When RAM_MODE=SDP, DOADO is the logical DO[31:0].
DOBDO[31:0]	Output	32	Port B data output bus/Data output bus addressed by RDADDR. When RAM_MODE=SDP, DOBDO is the logical DO[63:32].
DOPADOP[3:0]	Output	4	Port A parity data output bus/Data parity output bus addressed by RDADDR. When RAM_MODE=SDP, DOPADOP is the logical DOP[3:0].
DOPBDOP[3:0]	Output	4	Port B parity data output bus/Data parity output bus addressed by RDADDR. When RAM_MODE=SDP, DOPBDOP is the logical DOP[7:4].
ECCPARITY[7:0]	Output	8	8-bit data generated by the ECC encoder used by the ECC decoder for memory error detection and correction. Not used if RAM_MODE=TDP.
ENARDEN	Input	1	Port A RAM enable/Read enable.
ENBWREN	Input	1	Port B RAM enable/Write enable.
INJECTDBITERR	Input	1	Inject a double bit error if ECC feature is used.
INJECTSBITERR	Input	1	Inject a single bit error if ECC feature is used.
RDADDRECC[8:0]	Output	9	9-bit ECC read address. Not used when RAM_MODE=TDP.
REGCEAREGCE	Input	1	Port A output register clock enable input/Output register clock enable input (valid only when DO_REG=1).
REGCEB	Input	1	Port B output register clock enable (valid only when DO_REG=1 and RAM_MODE=TDP).

Port	Direction	Width	Function
RSTRAMARSTRAM	Input	1	Synchronous data latch set/reset to value indicated by SRVAL_A. RSTRAMARSTRAM sets/resets the BRAM data output latch when DO_REG=0 or 1. If DO_REG=1 there is a cycle of latency between the internal data latch node that is reset by RSTRAMARSTRAM and the DO output of the BRAM. This signal is RSTRAMA on port A when RAM_MODE=TDP and RSTRAM when RAM_MODE=SDP.
RSTRAMB	Input	1	Synchronous data latch set/reset to value indicated by SRVAL_B. RSTRAMB sets/resets the BRAM data output latch when DO_REG=0 or 1. If DO_REG=1 there is a cycle of latency between the internal data latch node that is reset by RSTRAMB and the DO output of the BRAM. Not used when RAM_MODE=SDP.
RSTREGARSTREG	Input	1	Synchronous output register set/reset to value indicated by SRVAL_A. RSTREGARSTREG sets/resets the output register when DO_REG=1. RSTREG_PRIORITY_A determines if this signal gets priority over REGCEAREGCE. This signal is RSTREGA on port A when RAM_MODE=TDP and RSTREG when RAM_MODE=SDP.
RSTREGB	Input	1	Synchronous output register set/reset to value indicated by SRVAL_B. RSTREGB sets/resets the output register when DO_REG=1. RSTREG_PRIORITY_B determines if this signal gets priority over REGCEB. Not used when RAM_MODE=SDP.
SBITERR	Output	1	Status output from ECC function to indicate a single bit error was detected. EN_ECC_READ needs to be TRUE in order to use this functionality. Not used when RAM_MODE=TDP.
WEA[3:0]	Input	4	Port A byte-wide write enable. Not used when RAM_MODE=SDP. See User Guide for WEA mapping for different port widths.
WEBWE[7:0]	Input	8	Port B byte-wide write enable/Write enable. See *Virtex®-6 Memory Resources User Guide* for WEBWE mapping for different port widths.

Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	Yes
Macro support	Yes

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
COLLISION CHECK	String	"ALL", "GENERATE_X_ ONLY", "NONE", "WARNING_ONLY"	"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

Attribute	Data Type	Allowed Values	Default	Description
				• "GENERATE_X_ONLY" - no warning, however affected outputs/memory go unknown (X) • "NONE"- no warning and affected outputs/memory retain last value **Note** Setting this to a value other than ALL can allow problems in the design to go unnoticed during simulation. Care should be taken when changing the value of this attribute.
DOA_REG	Integer	0, 1	0	A value of 1 enables 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 result in slower clock-to-out timing. Applies to port A in TDP mode and up to 36 lower bits (including parity bits) in SDP mode.
DOB_REG	Integer	0, 1	0	A value of 1 enables 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 result in slower clock-to-out timing. Applies to port B in TDP mode and upper bits (including parity bits) in SDP mode.
EN_ECC_READ	Boolean	FALSE, TRUE	FALSE	Enable the ECC decoder circuitry.
EN_ECC_WRITE	Boolean	FALSE, TRUE	FALSE	Enable the ECC encoder circuitry.
INIT_A	Hexa-decimal	Any 36 bit value	All zeros	Specifies the initial value on the Port A output after configuration. Applies to port A in TDP mode and up to 36 lower bits (including parity bits) in SDP mode.
INIT_B	Hexa-decimal	Any 36 bit value	All zeros	Specifies the initial value on the Port B output after configuration. Applies to port B in TDP mode and upper bits (including parity bits) in SDP mode.
INIT_FILE	String	String representing file name and location	NONE	File name of file used to specify initial RAM contents.
INIT_00 to INIT_7F	Hexa-decimal	All zeros to all ones	All zeros	Allows specification of the initial contents of the 32 kb data memory array.
INITP_00 to INITP_0F	Hexa-decimal	All zeros to all ones	All zeros	Allows specification of the initial contents of the 4 kb parity data memory array.

Attribute	Data Type	Allowed Values	Default	Description
RAM_EXTENSION_A	String	"NONE", "LOWER", "UPPER"	"NONE"	Selects port A cascade mode. If not cascading two block RAMs to form a 72K x 1 RAM, set to "NONE". If cascading RAMs, set to either "UPPER" or "LOWER" to indicate relative RAM location for proper configuration of the RAM. Not used if RAM_MODE=SDP.
RAM_EXTENSION_B	String	"NONE", "LOWER", "UPPER"	"NONE"	Selects port B cascade mode. If not cascading two block RAMs to form a 72K x 1 RAM, set to "NONE". If cascading RAMs, set to either "UPPER" or "LOWER" to indicate relative RAM location for proper configuration of the RAM. Not used if RAM_MODE=SDP.
RAM_MODE	String	"TDP", "SDP"	"TDP"	Selects simple dual port (SDP) or true dual port (TDP) mode.
READ_WIDTH_A	Integer	0, 1, 2, 4, 9, 18, 36, 72	0	Specifies the desired data width for a read on port A, including parity bits. This value must be 0 if the port is not used. Otherwise, it should be set to the desired port width.
READ_WIDTH_B	Integer	0, 1, 2, 4, 9, 18, 36, 72	0	Specifies the desired data width for a read on port B, including parity bits. This value must be 0 if the port is not used. Otherwise, it should be set to the desired port width.
RSTREG_PRIORITY_A	String	"RSTREG", "REGCE"	"RSTREG"	Selects register priority for RSTREG or REGCE. Applies to port A in TDP mode and up to 36 lower bits (including parity bits) in SDP mode.
RSTREG_PRIORITY_B	String	"RSTREG", "REGCE"	"RSTREG"	Selects register priority for RSTREG or REGCE. Applies to port B in TDP mode and upper bits (including parity bits) in SDP mode.
SRVAL_A	Hexa- decimal	Any 36 bit Value	All zeros	Specifies the output value of the RAM upon assertion of the synchronous reset (RSTREG) signal.
SRVAL_B	Hexa- decimal	Any 36 bit Value	All zeros	Specifies the output value of the RAM upon assertion of the synchronous reset (RSTREG) signal.
WRITEMODE	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.

Attribute	Data Type	Allowed Values	Default	Description
WRITE_WIDTH_A	Integer	0, 1, 2, 4, 9, 18, 36	0	Specifies the desired data width for a write on port A, including parity bits. This value must be 0 if the port is not used. Otherwise, it should be set to the desired port width.
WRITE_WIDTH_B	Integer	0, 1, 2, 4, 9, 18, 36, 72	0	Specifies the desired data width for a write on port B, including parity bits. This value must be 0 if the port is not used. Otherwise, it should be set to the desired port 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;


-- RAMB36E1: 36K-bit Configurable Synchronous Block RAM
--           Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

RAMB36E1_inst : RAMB36E1
generic map (
   -- Colision check: Values ("ALL", "WARNING_ONLY", "GENERATE_X_ONLY" or "NONE")
   SIM_COLLISION_CHECK => "ALL",
   -- DOA_REG, DOB_REG: Optional output register (0 or 1)
   DOA_REG => 0,
   DOB_REG => 0,
   -- Error Correction Circuitry (ECC): Encoder/decoder enable (TRUE/FALSE)
   EN_ECC_READ => FALSE,
   EN_ECC_WRITE => FALSE,
   -- INITP_00 to INITP_0F: Initial contents of the parity memory array
   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",
   INITP_08 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_09 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_0A => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_0B => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_0C => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_0D => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_0E => X"0000000000000000000000000000000000000000000000000000000000000000",
   INITP_0F => X"0000000000000000000000000000000000000000000000000000000000000000",
   -- INIT_00 to INIT_7F: Initial contents of the data memory array
   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_40 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_41 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_42 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_43 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_44 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_45 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_46 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_47 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_48 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_49 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_4A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_4B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_4C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_4D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_4E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_4F => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_50 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_51 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_52 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_53 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_54 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_55 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_56 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_57 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_58 => X"0000000000000000000000000000000000000000000000000000000000000000",
```

```
   INIT_59 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_5A => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_5B => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_5C => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_5D => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_5E => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_5F => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_60 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_61 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_62 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_63 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_64 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_65 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_66 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_67 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_68 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_69 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_6A => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_6B => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_6C => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_6D => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_6E => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_6F => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_70 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_71 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_72 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_73 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_74 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_75 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_76 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_77 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_78 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_79 => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_7A => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_7B => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_7C => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_7D => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_7E => X"0000000000000000000000000000000000000000000000000000000000000000",
   INIT_7F => X"0000000000000000000000000000000000000000000000000000000000000000",
   -- INIT_A, INIT_B: Initial values on output ports
   INIT_A => X"000000000",
   INIT_B => X"000000000",
   INIT_FILE => "NONE",                                                      -- RAM initialization
                                                                             -- file
   -- RAM_EXTENSION_A, RAM_EXTENSION_B: Selects cascade mode ("UPPER", "LOWER", or "NONE")
   RAM_EXTENSION_A => "NONE",
   RAM_EXTENSION_B => "NONE",
   RAM_MODE => "TDP",                                                        -- "SDP" or "TDP"
   RDADDR_COLLISION_HWCONFIG => "DELAYED_WRITE",                             -- "PERFORMANCE" or
                                                                             -- "DELAYED_WRITE"
   -- READ_WIDTH_A/B, WRITE_WIDTH_A/B: Read/write width per port
   READ_WIDTH_A => 0,                                                        -- 0, 1, 2, 4, 9, 18,
                                                                             -- 36, or 72
   READ_WIDTH_B => 0,                                                        -- 0, 1, 2, 4, 9, 18, or
                                                                             -- 36
   WRITE_WIDTH_A => 0,                                                       -- 0, 1, 2, 4, 9, 18, or
                                                                             -- 36
   WRITE_WIDTH_B => 0,                                                       -- 0, 1, 2, 4, 9, 18,
                                                                             -- 36, or 72
   -- RSTREG_PRIORITY_A, RSTREG_PRIORITY_B: Reset or enable priority ("RSTREG" or "REGCE")
   RSTREG_PRIORITY_A => "RSTREG",
   RSTREG_PRIORITY_B => "RSTREG",
   -- SRVAL_A, SRVAL_B: Set/reset value for output
   SRVAL_A => X"000000000",
   SRVAL_B => X"000000000",
   -- WriteMode: Value on output upon a write ("WRITE_FIRST", "READ_FIRST", or "NO_CHANGE")
   WRITE_MODE_A => "WRITE_FIRST",
   WRITE_MODE_B => "WRITE_FIRST"
)
port map (
   -- Cascade Signals: 1-bit (each) output: BRAM cascade ports (to create 64kx1)
   CASCADEOUTA => CASCADEOUTA,     -- 1-bit output: A port cascade output
   CASCADEOUTB => CASCADEOUTB,     -- 1-bit output: B port cascade output
```

```
   -- ECC Signals: 1-bit (each) output: Error Correction Circuitry ports
   DBITERR => DBITERR,              -- 1-bit output: double bit error status output
   ECCPARITY => ECCPARITY,          -- 8-bit output: generated error correction parity
   RDADDRECC => RDADDRECC,          -- 9-bit output: ECC read address
   SBITERR => SBITERR,              -- 1-bit output: Single bit error status output
   -- Port A Data: 32-bit (each) output: Port A data
   DOADO => DOADO,                  -- 32-bit output: A port data/LSB data output
   DOPADOP => DOPADOP,              -- 4-bit output: A port parity/LSB parity output
   -- Port B Data: 32-bit (each) output: Port B data
   DOBDO => DOBDO,                  -- 32-bit output: B port data/MSB data output
   DOPBDOP => DOPBDOP,              -- 4-bit output: B port parity/MSB parity output
   -- Cascade Signals: 1-bit (each) input: BRAM cascade ports (to create 64kx1)
   CASCADEINA => CASCADEINA,        -- 1-bit input: A port cascade input
   CASCADEINB => CASCADEINB,        -- 1-bit input: B port cascade input
   -- ECC Signals: 1-bit (each) input: Error Correction Circuitry ports
   INJECTDBITERR => INJECTDBITERR, -- 1-bit input: Inject a double bit error
   INJECTSBITERR => INJECTSBITERR, -- 1-bit input: Inject a single bit error
   -- Port A Address/Control Signals: 16-bit (each) input: Port A address and control signals (read port
   -- when RAM_MODE="SDP")
   ADDRARDADDR => ADDRARDADDR,      -- 16-bit input: A port address/Read address input
   CLKARDCLK => CLKARDCLK,          -- 1-bit input: A port clock/Read clock input
   ENARDEN => ENARDEN,              -- 1-bit input: A port enable/Read enable input
   REGCEAREGCE => REGCEAREGCE,      -- 1-bit input: A port register enable/Register enable input
   RSTRAMARSTRAM => RSTRAMARSTRAM, -- 1-bit input: A port set/reset input
   RSTREGARSTREG => RSTREGARSTREG, -- 1-bit input: A port register set/reset input
   WEA => WEA,                      -- 4-bit input: A port write enable input
   -- Port A Data: 32-bit (each) input: Port A data
   DIADI => DIADI,                  -- 32-bit input: A port data/LSB data input
   DIPADIP => DIPADIP,              -- 4-bit input: A port parity/LSB parity input
   -- Port B Address/Control Signals: 16-bit (each) input: Port B address and control signals (write port
   -- when RAM_MODE="SDP")
   ADDRBWRADDR => ADDRBWRADDR,      -- 16-bit input: B port address/Write address input
   CLKBWRCLK => CLKBWRCLK,          -- 1-bit input: B port clock/Write clock input
   ENBWREN => ENBWREN,              -- 1-bit input: B port enable/Write enable input
   REGCEB => REGCEB,                -- 1-bit input: B port register enable input
   RSTRAMB => RSTRAMB,              -- 1-bit input: B port set/reset input
   RSTREGB => RSTREGB,              -- 1-bit input: B port register set/reset input
   WEBWE => WEBWE,                  -- 8-bit input: B port write enable/Write enable input
   -- Port B Data: 32-bit (each) input: Port B data
   DIBDI => DIBDI,                  -- 32-bit input: B port data/MSB data input
   DIPBDIP => DIPBDIP               -- 4-bit input: B port parity/MSB parity input
);

-- End of RAMB36E1_inst instantiation
```

Verilog Instantiation Template

```
// RAMB36E1: 36K-bit Configurable Synchronous Block RAM
//           Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

RAMB36E1 #(
   // Colision check: Values ("ALL", "WARNING_ONLY", "GENERATE_X_ONLY" or "NONE")
   .SIM_COLLISION_CHECK("ALL"),
   // DOA_REG, DOB_REG: Optional output register (0 or 1)
   .DOA_REG(0),
   .DOB_REG(0),
   // Error Correction Circuitry (ECC): Encoder/decoder enable (TRUE/FALSE)
   .EN_ECC_READ("FALSE"),
   .EN_ECC_WRITE("FALSE"),
   // INITP_00 to INITP_0F: Initial contents of the parity memory array
   .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),
   .INITP_08(256'h0000000000000000000000000000000000000000000000000000000000000000),
   .INITP_09(256'h0000000000000000000000000000000000000000000000000000000000000000),
```

```
.INITP_0A(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_0B(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_0C(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_0D(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_0E(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_0F(256'h0000000000000000000000000000000000000000000000000000000000000000),
// INIT_00 to INIT_7F: Initial contents of the data memory array
.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_40(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_41(256'h0000000000000000000000000000000000000000000000000000000000000000),
```

```
.INIT_42(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_43(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_44(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_45(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_46(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_47(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_48(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_49(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_4A(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_4B(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_4C(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_4D(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_4E(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_4F(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_50(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_51(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_52(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_53(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_54(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_55(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_56(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_57(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_58(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_59(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_5A(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_5B(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_5C(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_5D(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_5E(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_5F(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_60(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_61(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_62(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_63(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_64(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_65(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_66(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_67(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_68(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_69(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_6A(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_6B(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_6C(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_6D(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_6E(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_6F(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_70(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_71(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_72(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_73(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_74(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_75(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_76(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_77(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_78(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_79(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_7A(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_7B(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_7C(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_7D(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_7E(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_7F(256'h0000000000000000000000000000000000000000000000000000000000000000),
// INIT_A, INIT_B: Initial values on output ports
.INIT_A(36'h000000000),
.INIT_B(36'h000000000),
.INIT_FILE("NONE"),                                                  // RAM initialization
                                                                     // file
// RAM_EXTENSION_A, RAM_EXTENSION_B: Selects cascade mode ("UPPER", "LOWER", or "NONE")
.RAM_EXTENSION_A("NONE"),
.RAM_EXTENSION_B("NONE"),
.RAM_MODE("TDP"),                                                    // "SDP" or "TDP"
.RDADDR_COLLISION_HWCONFIG("DELAYED_WRITE"),                         // "PERFORMANCE" or
                                                                     // "DELAYED_WRITE"
```

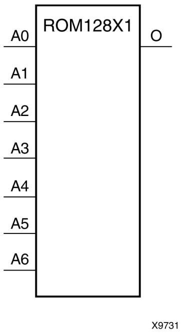
```
   // READ_WIDTH_A/B, WRITE_WIDTH_A/B: Read/write width per port
   .READ_WIDTH_A(0),                                                                  // 0, 1, 2, 4, 9, 18,
                                                                                      // 36, or 72
   .READ_WIDTH_B(0),                                                                  // 0, 1, 2, 4, 9, 18,
                                                                                      // or 36
   .WRITE_WIDTH_A(0),                                                                 // 0, 1, 2, 4, 9, 18,
                                                                                      // or 36
   .WRITE_WIDTH_B(0),                                                                 // 0, 1, 2, 4, 9, 18,
                                                                                      // 36, or 72
   // RSTREG_PRIORITY_A, RSTREG_PRIORITY_B: Reset or enable priority ("RSTREG" or "REGCE")
   .RSTREG_PRIORITY_A("RSTREG"),
   .RSTREG_PRIORITY_B("RSTREG"),
   // SRVAL_A, SRVAL_B: Set/reset value for output
   .SRVAL_A(36'h000000000),
   .SRVAL_B(36'h000000000),
   // WriteMode: Value on output upon a write ("WRITE_FIRST", "READ_FIRST", or "NO_CHANGE")
   .WRITE_MODE_A("WRITE_FIRST"),
   .WRITE_MODE_B("WRITE_FIRST")
)
RAMB36E1_inst (
   // Cascade Signals: 1-bit (each) output: BRAM cascade ports (to create 64kx1)
   .CASCADEOUTA(CASCADEOUTA),     // 1-bit output: A port cascade output
   .CASCADEOUTB(CASCADEOUTB),     // 1-bit output: B port cascade output
   // ECC Signals: 1-bit (each) output: Error Correction Circuitry ports
   .DBITERR(DBITERR),             // 1-bit output: double bit error status output
   .ECCPARITY(ECCPARITY),         // 8-bit output: generated error correction parity
   .RDADDRECC(RDADDRECC),         // 9-bit output: ECC read address
   .SBITERR(SBITERR),             // 1-bit output: Single bit error status output
   // Port A Data: 32-bit (each) output: Port A data
   .DOADO(DOADO),                 // 32-bit output: A port data/LSB data output
   .DOPADOP(DOPADOP),             // 4-bit output: A port parity/LSB parity output
   // Port B Data: 32-bit (each) output: Port B data
   .DOBDO(DOBDO),                 // 32-bit output: B port data/MSB data output
   .DOPBDOP(DOPBDOP),             // 4-bit output: B port parity/MSB parity output
   // Cascade Signals: 1-bit (each) input: BRAM cascade ports (to create 64kx1)
   .CASCADEINA(CASCADEINA),       // 1-bit input: A port cascade input
   .CASCADEINB(CASCADEINB),       // 1-bit input: B port cascade input
   // ECC Signals: 1-bit (each) input: Error Correction Circuitry ports
   .INJECTDBITERR(INJECTDBITERR), // 1-bit input: Inject a double bit error
   .INJECTSBITERR(INJECTSBITERR), // 1-bit input: Inject a single bit error
   // Port A Address/Control Signals: 16-bit (each) input: Port A address and control signals (read port
   // when RAM_MODE="SDP")
   .ADDRARDADDR(ADDRARDADDR),     // 16-bit input: A port address/Read address input
   .CLKARDCLK(CLKARDCLK),         // 1-bit input: A port clock/Read clock input
   .ENARDEN(ENARDEN),             // 1-bit input: A port enable/Read enable input
   .REGCEAREGCE(REGCEAREGCE),     // 1-bit input: A port register enable/Register enable input
   .RSTRAMARSTRAM(RSTRAMARSTRAM), // 1-bit input: A port set/reset input
   .RSTREGARSTREG(RSTREGARSTREG), // 1-bit input: A port register set/reset input
   .WEA(WEA),                     // 4-bit input: A port write enable input
   // Port A Data: 32-bit (each) input: Port A data
   .DIADI(DIADI),                 // 32-bit input: A port data/LSB data input
   .DIPADIP(DIPADIP),             // 4-bit input: A port parity/LSB parity input
   // Port B Address/Control Signals: 16-bit (each) input: Port B address and control signals (write port
   // when RAM_MODE="SDP")
   .ADDRBWRADDR(ADDRBWRADDR),     // 16-bit input: B port address/Write address input
   .CLKBWRCLK(CLKBWRCLK),         // 1-bit input: B port clock/Write clock input
   .ENBWREN(ENBWREN),             // 1-bit input: B port enable/Write enable input
   .REGCEB(REGCEB),               // 1-bit input: B port register enable input
   .RSTRAMB(RSTRAMB),             // 1-bit input: B port set/reset input
   .RSTREGB(RSTREGB),             // 1-bit input: B port register set/reset input
   .WEBWE(WEBWE),                 // 8-bit input: B port write enable/Write enable input
   // Port B Data: 32-bit (each) input: Port B data
   .DIBDI(DIBDI),                 // 32-bit input: B port data/MSB data input
   .DIPBDIP(DIPBDIP)              // 4-bit input: B port parity/MSB parity input
);

// End of RAMB36E1_inst instantiation
```

For More Information

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

ROM128X1

Primitive: 128-Deep by 1-Wide ROM



Introduction

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

Logic Table

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

Design Entry Method

Instantiation	Yes
Inference	Recommended
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
--           Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

-- End of ROM128X1_inst instantiation
```

Verilog Instantiation Template

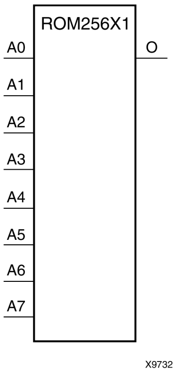
```
// ROM128X1: 128 x 1 Asynchronous Distributed (LUT) ROM
//           Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

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

// End of ROM128X1_inst instantiation
```

For More Information

See the *Virtex-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
--           Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

-- End of ROM256X1_inst instantiation
```

Verilog Instantiation Template

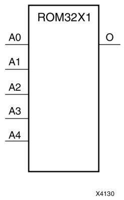
```
// ROM256X1: 256 x 1 Asynchronous Distributed (LUT) ROM
//           Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

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

// End of ROM256X1_inst instantiation
```

For More Information

See the *Virtex-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
--          Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

ROM32X1_inst : ROM32X1
generic map (
   INIT => X"00000000")
port map (
   O => O,    -- ROM output
   A0 => A0, -- ROM address[0]
   A1 => A1, -- ROM address[1]
   A2 => A2, -- ROM address[2]
   A3 => A3, -- ROM address[3]
   A4 => A4  -- ROM address[4]
);
-- End of ROM32X1_inst instantiation
```

Verilog Instantiation Template

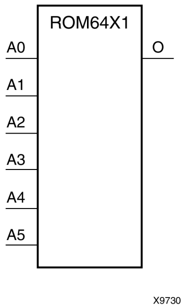
```
// ROM32X1: 32 x 1 Asynchronous Distributed (LUT) ROM
//          Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

ROM32X1 #(
   .INIT(32'h00000000) // Contents of ROM
) ROM32X1_inst (
   .O(O),    // ROM output
   .A0(A0), // ROM address[0]
   .A1(A1), // ROM address[1]
   .A2(A2), // ROM address[2]
   .A3(A3), // ROM address[3]
   .A4(A4)  // ROM address[4]
);

// End of ROM32X1_inst instantiation
```

For More Information

See the *Virtex-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
--          Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

-- End of ROM64X1_inst instantiation
```

Verilog Instantiation Template

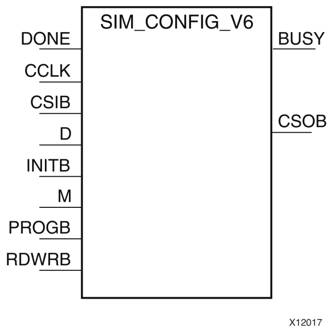
```
// ROM64X1: 64 x 1 Asynchronous Distributed (LUT) ROM
//          Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

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

// End of ROM64X1_inst instantiation
```

For More Information

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

SIM_CONFIG_V6

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_VIRTEX6 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_V6: Behavioral Simulation-only Model of FPGA SelectMap Configuration
--               Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

SIM_CONFIG_V6_inst : SIM_CONFIG_V6
generic map (
   ICAP_SUPPORT => FALSE,      -- Using ICAP, TRUE or FALSE
   ICAP_WIDTH => "X8",         -- ICAP width, "X8", "X16", "X32"
                               -- Do not need to change/specify if
                               -- ICAP_SUPPORT-FALSE
   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_V6_inst instantiation
```

Verilog Instantiation Template

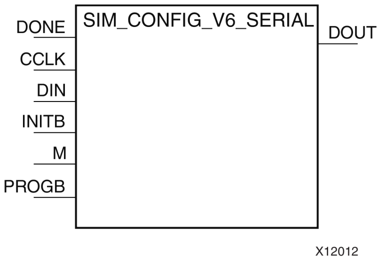
```
// SIM_CONFIG_V6: Behavioral Simulation-only Model of FPGA SelectMap Configuration
//                Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

SIM_CONFIG_V6 #(
   .DEVICE_ID(32'h00000000), // Specify DEVICE_ID
   .ICAP_SUPPORT("FALSE"),   // Using ICAP, "TRUE" or "FALSE"
   .ICAP_WIDTH("X8")         // ICAP width, "X8", "X16", "X32"
                             // Do not need to change/specify if ICAP_SUPPORT="FALSE"
) SIM_CONFIG_V6_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
   .CSB(CSB),   // 1-bit input chip select
   .D(D),       // 32-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_V6_inst instantiation
```

For More Information

- See the *Synthesis and Simulation Design Guide (UG626)*.
- See the *Virtex-6 FPGA User Documentation (User Guides and Data Sheets)*.

SIM_CONFIG_V6_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_V6_SERIAL: Behavioral Simulation-only Model of FPGA Serial Configuration
--                       Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

SIM_CONFIG_V6_SERIAL_inst : SIM_CONFIG_V6_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_V6_SERIAL_inst instantiation
```

Verilog Instantiation Template

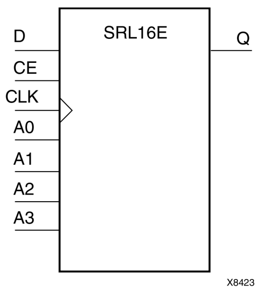
```
// SIM_CONFIG_V6_SERIAL: Behavioral Simulation-only Model of FPGA Serial Configuration
//                       Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

SIM_CONFIG_V6_SERIAL #(
   .DEVICE_ID(32'h00000000)  // Specify DEVICE_ID
) SIM_CONFIG_V6_SERIAL_inst (
   .DONE(DONE),     // 1-bit bi-directional Done pin
   .DOUT(DOUT),     // 1-bit data output 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),           // 3-bit input Mode pins
   .PROGB(PROGB)    // 1-bit input Program pin
);

// End of SIM_CONFIG_V6_SERIAL_inst instantiation
```

For More Information

- See the *Synthesis and Simulation Design Guide (UG626)*.
- See the *Virtex-6 FPGA User Documentation (User Guides and Data Sheets)*.

SRL16E

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



Introduction

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

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

- **To create a fixed-length shift register -**Drive the A3 through A0 inputs with static values. The length of the shift register can vary from 1 bit to 16 bits, as determined by the following formula: Length = (8 x A3) +(4 x A2) + (2 x A1) + A0 +1 If A3, A2, A1, and A0 are all zeros (0000), the shift register is one bit long. If they are all ones (1111), it is 16 bits long.
- **To change the length of the shift register dynamically -**Change the values driving the A3 through A0 inputs. For example, if A2, A1, and A0 are all ones (111) and A3 toggles between a one (1) and a zero (0), the length of the shift register changes from 16 bits to 8 bits. Internally, the length of the shift register is always 16 bits and the input lines A3 through A0 select which of the 16 bits reach the output.

The shift register LUT contents are initialized by assigning a four-digit hexadecimal number to an INIT attribute. The first, or the left-most, hexadecimal digit is the most significant bit. If an INIT value is not specified, it defaults to a value of four zeros (0000) so that the shift register LUT is cleared during configuration.

When CE is High, the data (D) is loaded into the first bit of the shift register during the Low-to-High clock (CLK) transition. During subsequent Low-to-High clock transitions, when CE is High, data shifts to the next highest bit position as new data is loaded. The data appears on the Q output when the shift register length determined by the address inputs is reached. When CE is Low, the register ignores clock transitions.

Logic Table

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

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
--        Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

SRL16E_inst : SRL16E
generic map (
   INIT => X"0000")
port map (
   Q => Q,       -- SRL data output
   A0 => A0,     -- Select[0] input
   A1 => A1,     -- Select[1] input
   A2 => A2,     -- Select[2] input
   A3 => A3,     -- Select[3] input
   CE => CE,     -- Clock enable input
   CLK => CLK,   -- Clock input
   D => D        -- SRL data input
);

-- End of SRL16E_inst instantiation
```

Verilog Instantiation Template

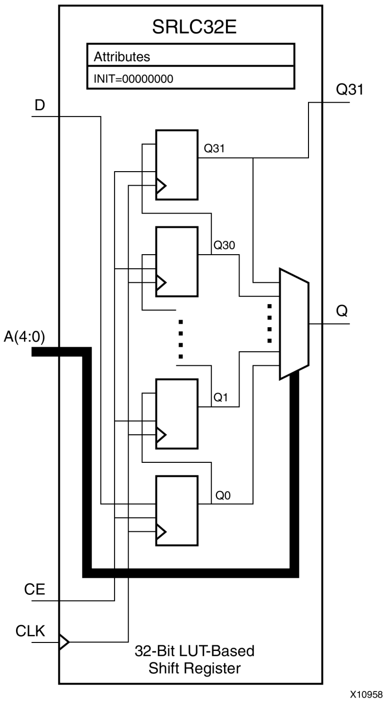
```
// SRL16E: 16-bit shift register LUT with clock enable operating on posedge of clock
//         Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

SRL16E #(
   .INIT(16'h0000) // Initial Value of Shift Register
) SRL16E_inst (
   .Q(Q),       // SRL data output
   .A0(A0),     // Select[0] input
   .A1(A1),     // Select[1] input
   .A2(A2),     // Select[2] input
   .A3(A3),     // Select[3] input
   .CE(CE),     // Clock enable input
   .CLK(CLK),   // Clock input
   .D(D)        // SRL data input
);

// End of SRL16E_inst instantiation
```

For More Information

See the *Virtex-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
--          Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

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

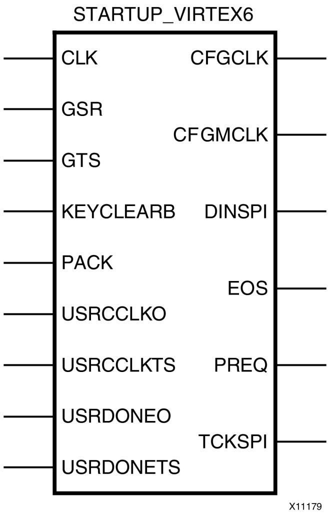
```
// SRLC32E: 32-bit variable length cascadable shift register LUT
//          with clock enable
//          Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

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

STARTUP_VIRTEX6

Primitive: Virtex®-6 Configuration Start-Up Sequence 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 of the device, and to access the configuration clock to the internal logic.

Port Descriptions

Port	Direction	Width	Function
CFGCLK	Output	1	Configuration main clock output. This pin is an output into the FPGA fabric. It outputs a clock signal with the frequency defined in the Bitgen options. Its source is the internal ring oscillator.
CFGMCLK	Output	1	Configuration internal oscillator clock output. This pin is an output into the FPGA fabric. It outputs a clock signal with a constant 50 MHz frequency. Its source is the internal ring oscillator.
CLK	Input	1	User startup clock. This pin is an input from the FPGA fabric. It drives the startup clock for the device startup sequence. It is essentially a user-defined CCLK.
DINSPI	Output	1	DIN SPI PROM access output. This pin is an output into the FPGA fabric. The data on this pin is the serial data being read from a SPI PROM connected to the FPGA. This pin is useful for reading back SPI PROM contents in order to perform a "verify" operation.
EOS	Output	1	Active High signal indicates the End Of Configuration. This pin is an output into the FPGA fabric. It echoes the "end of startup" flag into the FPGA fabric. This pin can be used as a reset signal.

Port	Direction	Width	Function
GSR	Input	1	Global Set/Reset (GSR) input (GSR cannot be used for the port name). This pin is an input from the FPGA fabric. It drives the state of the Global Set/Reset (GSR) pin manually. For most applications, this should be tied low.
GTS	Input	1	Global Tristate (GTS) input (GTS cannot be used for the port name). This pin is an input from the FPGA fabric. It drives the state of the Global Tristate (GTS) pin manually. For most applications, this should be tied low.
KEYCLEARB	Input	1	Clear AES Decrypter Key from Battery-Backed RAM (BBRAM). This pin is an input from the FPGA fabric. When held low for 200 ns, it erases the contents of the decryption keys from the BBRAM. This pin is only enabled if the PROG_USR attribute is set. It can be tied high for "safe" operations.
PACK	Input	1	PROGRAM acknowledge input. This pin is an input from the FPGA fabric. It "acknowledges" the assertion of the PROG_B signal and allows the remainder of the PROG_B state machine to continue resetting the FPGA. This pin is only enabled if PROG_USR attribute is set. It can be tied low for "safe" operations.
PREQ	Output	1	PROGRAM request to fabric output. This pin is an output into the FPGA fabric. This pin is the "request" from the PROG_B state machine to reset the device. This allows the assertion of the PROG_B request to be gated until the design is in a state where the reset can be completed. This pin is only enabled if PROG_USR attribute is set. It can be left open/floating for "safe" operation.
TCKSPI	Output	1	TCK configuration pin access output. This pin is an output into the FPGA fabric. It is a direct echo of the CCLK clock being driven onto the FPGA's configuration interface. This pin is useful for synchronizing an internal state machine to CCLK.
USRCCLKO	Input	1	User CCLK input. This pin is an input from the FPGA fabric. It drives a custom, fabric-generated clock frequency onto CCLK at the FPGA pin. This is useful for post-configuration access of external PROMs (notably SPI PROMs).
USRCCLKTS	Input	1	Internal user CCLK 3-state enable. This pin is an input from the FPGA fabric. It enables the tristate nature of the FPGA's CCLK pin. Generally, this should be tied low to prevent tri-stating of the CCLK pin.
USRDONEO	Input	1	Internal user DONE pin output control. This pin is an input from the FPGA fabric. It directly drives the FPGA DONE pin.
USRDONETS	Input	1	User DONE 3-state enable. This pin is an input from the FPGA fabric. It enables the tristate nature of the FPGA's DONE pin. Generally, this should be tied low. Tying this high will inhibit the assertion of DONE.

Design Entry Method

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

If the dedicated global tristate is to be 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. CFGMCLK and CFGCLK allow access to the internal configuration clocks, while EOS signals the end of the configuration startup sequence.

If you are configuring the device using a SPI PROM, and access to the SPI PROM is necessary after configuration, use the USRCCLKO and DINSPI pins of the component to gain access to the otherwise dedicated configuration input pins.

Available Attributes

Attribute	Data Type	Allowed Values	Default	Description
PROG_USR	Boolean	FALSE, TRUE	FALSE	Activate program event security feature. Can only be used when an encrypted bitstream is in use.

VHDL 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_VIRTEX6: STARTUP Block
--                  Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

STARTUP_VIRTEX6_inst : STARTUP_VIRTEX6
generic map (
   PROG_USR => FALSE  -- Activate program event security feature. Requires encrypted bitstreams.
)
port map (
   CFGCLK => CFGCLK,       -- 1-bit output: Configuration main clock output
   CFGMCLK => CFGMCLK,     -- 1-bit output: Configuration internal oscillator clock output
   DINSPI => DINSPI,       -- 1-bit output: DIN SPI PROM access output
   EOS => EOS,             -- 1-bit output: Active high output signal indicating the End Of Configuration.
   PREQ => PREQ,           -- 1-bit output: PROGRAM request to fabric output
   TCKSPI => TCKSPI,       -- 1-bit output: TCK configuration pin access output
   CLK => CLK,             -- 1-bit input: User start-up 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)
   PACK => PACK,           -- 1-bit input: PROGRAM acknowledge input
   USRCCLKO => USRCCLKO,   -- 1-bit input: User CCLK input
   USRCCLKTS => USRCCLKTS, -- 1-bit input: User CCLK 3-state enable input
   USRDONEO => USRDONEO,   -- 1-bit input: User DONE pin output control
   USRDONETS => USRDONETS  -- 1-bit input: User DONE 3-state enable output
);

-- End of STARTUP_VIRTEX6_inst instantiation
```

Verilog Instantiation Template

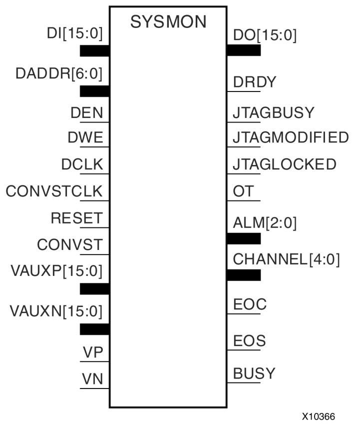
```
// STARTUP_VIRTEX6: STARTUP Block
//                  Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

STARTUP_VIRTEX6 #(
   .PROG_USR("FALSE")  // Activate program event security feature. Requires encrypted bitstreams.
)
STARTUP_VIRTEX6_inst (
   .CFGCLK(CFGCLK),       // 1-bit output: Configuration main clock output
   .CFGMCLK(CFGMCLK),     // 1-bit output: Configuration internal oscillator clock output
   .DINSPI(DINSPI),       // 1-bit output: DIN SPI PROM access output
   .EOS(EOS),             // 1-bit output: Active high output signal indicating the End Of Configuration.
   .PREQ(PREQ),           // 1-bit output: PROGRAM request to fabric output
   .TCKSPI(TCKSPI),       // 1-bit output: TCK configuration pin access output
   .CLK(CLK),             // 1-bit input: User start-up 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)
   .PACK(PACK),           // 1-bit input: PROGRAM acknowledge input
   .USRCCLKO(USRCCLKO),   // 1-bit input: User CCLK input
   .USRCCLKTS(USRCCLKTS), // 1-bit input: User CCLK 3-state enable input
   .USRDONEO(USRDONEO),   // 1-bit input: User DONE pin output control
   .USRDONETS(USRDONETS)  // 1-bit input: User DONE 3-state enable output
);

// End of STARTUP_VIRTEX6_inst instantiation
```

For More Information

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

SYSMON

Primitive: System Monitor



Introduction

This design element is built around a 10-bit, 200-kSPS (kilosamples per second) Analog-to-Digital Converter (ADC). When combined with a number of on-chip sensors, the ADC is used to measure FPGA physical operating parameters, including on-chip power supply voltages and die temperatures. Access to external voltages is provided through a dedicated analog-input pair (VP/VN) and 16 user-selectable analog inputs, known as auxiliary analog inputs (VAUXP[15:0], VAUXN[15:0]). The external analog inputs allow the ADC to monitor the physical environment of the board or enclosure.

Port Descriptions

Port	Direction	Width	Function
ALM[2:0]	Output	3	3-bit output alarm for temp, Vccint and Vccaux
BUSY	Output	1	1-bit output ADC busy signal
CHANNEL[4:0]	Output	5	5-bit output channel selection
CONVST	Input	1	1-bit input convert start
CONVSTCLK	Input	1	1-bit input convert start clock
DADDR[6:0]	Input	7	7-bit input address bus for dynamic reconfig
DCLK	Input	1	1-bit input clock for dynamic reconfig
DEN	Input	1	1-bit input enable for dynamic reconfig
DI[15:0]	Input	16	16-bit input data bus for dynamic reconfig
DO[15:0]	Output	16	16-bit output data bus for dynamic reconfig
DRDY	Output	1	1-bit output data ready for dynamic reconfig
DWE	Input	1	1-bit input write enable for dynamic reconfig
EOC	Output	1	1-bit output end of conversion
EOS	Output	1	1-bit output end of sequence
JTAGBUSY	Output	1	1-bit output JTAG DRP busy
JTAGLOCKED	Output	1	1-bit output DRP port lock
JTAGMODIFIED	Output	1	1-bit output JTAG write to DRP

Port	Direction	Width	Function
OT	Output	1	1-bit output over temperature alarm
RESET	Input	1	1-bit input active high reset
VAUXN[15:0]	Input	16	16-bit input N-side auxiliary analog input
VAUXP[15:0]	Input	16	16-bit input P-side auxiliary analog input
VN	Input	1	1-bit input N-side analog input
VP	Input	1	1-bit input P-side analog input

Design Entry Method

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

Connect all desired input and output ports and set the appropriate attributes for the desired behavior of this component. For simulation, provide a text file to give the analog and temperature to the model. The format for this file is as follows:

```
// Must use valid headers on all columns
// Comments can be added to the stimulus file using '//'
TIME TEMP VCCAUX VCCINT VP VN VAUXP[0] VAUXN[0]
00000 45 2.5 1.0 0.5 0.0 0.7 0.0
05000 85 2.45 1.1 0.3 0.0 0.2 0.0
// Time stamp data is in nano seconds (ns)
// Temperature is recorded in C (degrees centigrade)
// All other channels are recorded as V (Volts)
// Valid column headers are:
// TIME, TEMP, VCCAUX, VCCINT, VP, VN,
// VAUXP[0], VAUXN[0],...............VAUXP[15], VAUXN[15]
// External analog inputs are differential so VP = 0.5 and VN = 0.0 the
// input on channel VP/VN is 0.5 - 0.0 = 0.5V
```

Note When compiling the included code, please do not add any extraneous spaces to the text as this could cause compilation to fail.

Available Attributes

Attribute	Data Type	Allowed_Values	Default	Description
INIT_40	Hexa-decimal	16'h0000 to 16'hffff	16'h0000	Configuration register 0
INIT_41	Hexa-decimal	16'h0000 to 16'hffff	16'h0000	Configuration register 1
INIT_42	Hexa-decimal	16'h0000 to 16'hffff	16'h0800	Configuration register 2
INIT_43	Hexa-decimal	16'h0000 to 16'hffff	16'h0000	Test register 0
INIT_44	Hexa-decimal	16'h0000 to 16'hffff	16'h0000	Test register 1
INIT_45	Hexa-decimal	16'h0000 to 16'hffff	16'h0000	Test register 2

Attribute	Data Type	Allowed_Values	Default	Description
INIT_46	Hexa-decimal	16'h0000 to 16'hffff	16'h0000	Test register 3
INIT_47	Hexa-decimal	16'h0000 to 16'hffff	16'h0000	Test register 4
INIT_48	Hexa-decimal	16'h0000 to 16'hffff	16'h0000	Sequence register 0
INIT_49	Hexa-decimal	16'h0000 to 16'hffff	16'h0000	Sequence register 1
INIT_4A	Hexa-decimal	16'h0000 to 16'hffff	16'h0000	Sequence register 2
INIT_4B	Hexa-decimal	16'h0000 to 16'hffff	16'h0000	Sequence register 3
INIT_4C	Hexa-decimal	16'h0000 to 16'hffff	16'h0000	Sequence register 4
INIT_4D	Hexa-decimal	16'h0000 to 16'hffff	16'h0000	Sequence register 5
INIT_4E	Hexa-decimal	16'h0000 to 16'hffff	16'h0000	Sequence register 6
INIT_4F	Hexa-decimal	16'h0000 to 16'hffff	16'h0000	Sequence register 7
INIT_50	Hexa-decimal	16'h0000 to 16'hffff	16'h0000	Alarm limit register 0
INIT_51	Hexa-decimal	16'h0000 to 16'hffff	16'h0000	Alarm limit register 1
INIT_52	Hexa-decimal	16'h0000 to 16'hffff	16'h0000	Alarm limit register 2
INIT_53	Hexa-decimal	16'h0000 to 16'hffff	16'h0000	Alarm limit register 3
INIT_54	Hexa-decimal	16'h0000 to 16'hffff	16'h0000	Alarm limit register 4
INIT_55	Hexa-decimal	16'h0000 to 16'hffff	16'h0000	Alarm limit register 5
INIT_56	Hexa-decimal	16'h0000 to 16'hffff	16'h0000	Alarm limit register 6
INIT_57	Hexa-decimal	16'h0000 to 16'hffff	16'h0000	Alarm limit register 7
SIM_DEVICE	String	"VIRTEX5", "VIRTEX6"	"VIRTEX5"	Specifies the target device family for simulation.
SIM_MONITOR_FILE	String	String representing file name and location	design.txt	Simulation analog entry file

VHDL Instantiation Template

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

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

```
-- SYSMON: System Monitor
--         Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

SYSMON_inst : SYSMON
generic map (
   -- INIT_40 - INIT_42: System Monitor configuration registers
   INIT_40 => X"0000",
   INIT_41 => X"0000",
   INIT_42 => X"0800",
   -- INIT_43 - INIT_47: System Monitor Test registers (do not edit)
   INIT_43 => X"0000",
   INIT_44 => X"0000",
   INIT_45 => X"0000",
   INIT_46 => X"0000",
   INIT_47 => X"0000",
   -- INIT_48 - INIT_4F: Sequence registers for the Channel Sequencer
   INIT_48 => X"0000",
   INIT_49 => X"0000",
   INIT_4A => X"0000",
   INIT_4B => X"0000",
   INIT_4C => X"0000",
   INIT_4D => X"0000",
   INIT_4E => X"0000",
   INIT_4F => X"0000",
   -- INIT_50 - INIT_57: Alarm threshold registers
   INIT_50 => X"0000",
   INIT_51 => X"0000",
   INIT_52 => X"0000",
   INIT_53 => X"0000",
   INIT_54 => X"0000",
   INIT_55 => X"0000",
   INIT_56 => X"0000",
   INIT_57 => X"0000",
   -- Simulation attributes: Set for proper simulation behavior
   SIM_DEVICE => "VIRTEX5",          -- Must be set to VIRTEX6
   SIM_MONITOR_FILE => "design.txt"  -- Analog simulation data file name
)
port map (
   -- Alarm Ports: 3-bit (each) output: ALM, OT
   ALM => ALM,                     -- 3-bit output: output alarm for temp, Vccint and Vccaux
   OT => OT,                       -- 1-bit output: Over-Temperature alarm output
   -- DRP Ports: 16-bit (each) output: Dynamic Reconfiguration Ports
   DO => DO,                       -- 16-bit output: DRP output data bus
   DRDY => DRDY,                   -- 1-bit output: DRP data ready output signal
   -- Status Ports: 1-bit (each) output: SYSMON status ports
   BUSY => BUSY,                   -- 1-bit output: ADC busy output
   CHANNEL => CHANNEL,             -- 5-bit output: Channel selection outputs
   EOC => EOC,                     -- 1-bit output: End of Conversion output
   EOS => EOS,                     -- 1-bit output: End of Sequence output
   JTAGBUSY => JTAGBUSY,           -- 1-bit output: JTAG DRP transaction in progress output
   JTAGLOCKED => JTAGLOCKED,       -- 1-bit output: JTAG requested DRP port lock output
   JTAGMODIFIED => JTAGMODIFIED, -- 1-bit output: JTAG Write to the DRP has occurred output
   -- Auxiliary Analog-Input Pairs: 16-bit (each) input: VAUXP[15:0], VAUXN[15:0]
   VAUXN => VAUXN,                 -- 16-bit input: N-side auxiliary analog input
   VAUXP => VAUXP,                 -- 16-bit input: P-side auxiliary analog input
   -- Control and Clock Ports: 1-bit (each) input: Reset and Converstion Start
   CONVST => CONVST,               -- 1-bit input: Convert start input
   CONVSTCLK => CONVSTCLK,         -- 1-bit input: Convert start input
   RESET => RESET,                 -- 1-bit input: Active-high reset input
   -- DRP Ports: 7-bit (each) input: Dynamic Reconfiguration Ports
   DADDR => DADDR,                 -- 7-bit input: DRP input address bus
   DCLK => DCLK,                   -- 1-bit input: DRP clock input
   DEN => DEN,                     -- 1-bit input: DRP input enable signal
   DI => DI,                       -- 16-bit input: DRP input data bus
   DWE => DWE,                     -- 1-bit input: DRP write enable input
   -- Dedicated Analog Input Pair: 1-bit (each) input: VP/VN
   VN => VN,                       -- 1-bit input: N-side analog input
   VP => VP                        -- 1-bit input: P-side analog input
);

-- End of SYSMON_inst instantiation
```

Verilog Instantiation Template

```
// SYSMON: System Monitor
//         Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

SYSMON #(
   // INIT_40 - INIT_42: System Monitor configuration registers
   .INIT_40(16'h0000),
   .INIT_41(16'h0000),
   .INIT_42(16'h0800),
   // INIT_43 - INIT_47: System Monitor Test registers (do not edit)
   .INIT_43(16'h0000),
   .INIT_44(16'h0000),
   .INIT_45(16'h0000),
   .INIT_46(16'h0000),
   .INIT_47(16'h0000),
   // INIT_48 - INIT_4F: Sequence registers for the Channel Sequencer
   .INIT_48(16'h0000),
   .INIT_49(16'h0000),
   .INIT_4A(16'h0000),
   .INIT_4B(16'h0000),
   .INIT_4C(16'h0000),
   .INIT_4D(16'h0000),
   .INIT_4E(16'h0000),
   .INIT_4F(16'h0000),
   // INIT_50 - INIT_57: Alarm threshold registers
   .INIT_50(16'h0000),
   .INIT_51(16'h0000),
   .INIT_52(16'h0000),
   .INIT_53(16'h0000),
   .INIT_54(16'h0000),
   .INIT_55(16'h0000),
   .INIT_56(16'h0000),
   .INIT_57(16'h0000),
   // Simulation attributes: Set for proper simulation behavior
   .SIM_DEVICE("VIRTEX5"),          // Must be set to VIRTEX6
   .SIM_MONITOR_FILE("design.txt")  // Analog simulation data file name
)
SYSMON_inst (
   // Alarm Ports: 3-bit (each) output: ALM, OT
   .ALM(ALM),                    // 3-bit output: output alarm for temp, Vccint and Vccaux
   .OT(OT),                      // 1-bit output: Over-Temperature alarm output
   // DRP Ports: 16-bit (each) output: Dynamic Reconfiguration Ports
   .DO(DO),                      // 16-bit output: DRP output data bus
   .DRDY(DRDY),                  // 1-bit output: DRP data ready output signal
   // Status Ports: 1-bit (each) output: SYSMON status ports
   .BUSY(BUSY),                  // 1-bit output: ADC busy output
   .CHANNEL(CHANNEL),            // 5-bit output: Channel selection outputs
   .EOC(EOC),                    // 1-bit output: End of Conversion output
   .EOS(EOS),                    // 1-bit output: End of Sequence output
   .JTAGBUSY(JTAGBUSY),          // 1-bit output: JTAG DRP transaction in progress output
   .JTAGLOCKED(JTAGLOCKED),      // 1-bit output: JTAG requested DRP port lock output
   .JTAGMODIFIED(JTAGMODIFIED), // 1-bit output: JTAG Write to the DRP has occurred output
   // Auxiliary Analog-Input Pairs: 16-bit (each) input: VAUXP[15:0], VAUXN[15:0]
   .VAUXN(VAUXN),                // 16-bit input: N-side auxiliary analog input
   .VAUXP(VAUXP),                // 16-bit input: P-side auxiliary analog input
   // Control and Clock Ports: 1-bit (each) input: Reset and Converstion Start
   .CONVST(CONVST),              // 1-bit input: Convert start input
   .CONVSTCLK(CONVSTCLK),        // 1-bit input: Convert start input
   .RESET(RESET),                // 1-bit input: Active-high reset input
   // DRP Ports: 7-bit (each) input: Dynamic Reconfiguration Ports
   .DADDR(DADDR),                // 7-bit input: DRP input address bus
   .DCLK(DCLK),                  // 1-bit input: DRP clock input
   .DEN(DEN),                    // 1-bit input: DRP input enable signal
   .DI(DI),                      // 16-bit input: DRP input data bus
   .DWE(DWE),                    // 1-bit input: DRP write enable input
   // Dedicated Analog Input Pair: 1-bit (each) input: VP/VN
   .VN(VN),                      // 1-bit input: N-side analog input
   .VP(VP)                       // 1-bit input: P-side analog input
```

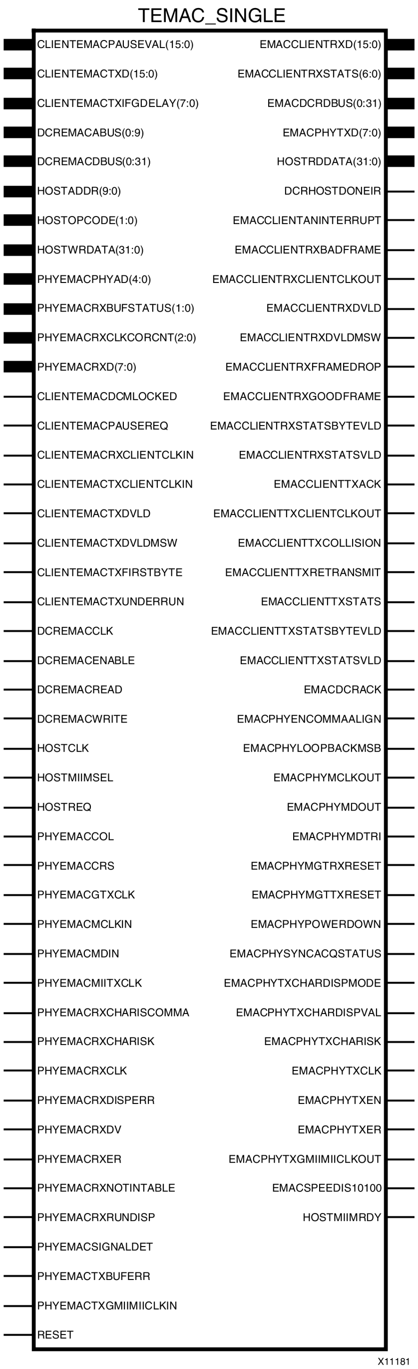
```
);
// End of SYSMON_inst instantiation
```

For More Information

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

TEMAC_SINGLE

Primitive: Tri-mode Ethernet Media Access Controller (MAC)



Introduction

The TEMAC_SINGLE library primitive provides the ports and attributes necessary to instantiate the Virtex®-6 FPGA Embedded Tri-Mode Ethernet MAC. Because it encompasses SecureIP encrypted HDL, it is also used for functional and timing simulations. This primitive can be simplified for specific customer needs by using the CORE Generator™ tool to create Ethernet MAC wrappers.

Design Entry Method

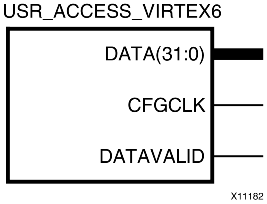
To instantiate this component, use the Embedded Development Kit (EDK) or an associated core containing the component. Xilinx does not recommend direct instantiation of this component.

For More Information

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

USR_ACCESS_VIRTEX6

Primitive: Virtex-6 User Access Register



Introduction

This design element enables access to a 32-bit register within the configuration logic. You will thus be able to read the data from the bitstream. One use for this component is to allow data stored in bitstream storage source to be accessed by the FPGA design after configuration.

Port Descriptions

Port	Direction	Width	Function
CFGCLK	Output	1	Configuration Clock output
DATA[31:0]	Output	32	Configuration Data output
DATAVALID	Output	1	Active high DATA port contains valid data

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;


-- USR_ACCESS_VIRTEX6: Configuration Data Access
--                     Virtex-6
-- Xilinx HDL Libraries Guide, version 14.2

USR_ACCESS_VIRTEX6_inst : USR_ACCESS_VIRTEX6
port map (
   CFGCLK => CFGCLK,        -- 1-bit output: Configuration Clock output
   DATA => DATA,            -- 32-bit output: Configuration Data output
   DATAVALID => DATAVALID  -- 1-bit output: Active high data valid output
);

-- End of USR_ACCESS_VIRTEX6_inst instantiation
```

Verilog Instantiation Template

```
// USR_ACCESS_VIRTEX6: Configuration Data Access
//                     Virtex-6
// Xilinx HDL Libraries Guide, version 14.2

USR_ACCESS_VIRTEX6 USR_ACCESS_VIRTEX6_inst (
   .CFGCLK(CFGCLK),       // 1-bit output: Configuration Clock output
   .DATA(DATA),           // 32-bit output: Configuration Data output
   .DATAVALID(DATAVALID)  // 1-bit output: Active high data valid output
);

// End of USR_ACCESS_VIRTEX6_inst instantiation
```

For More Information

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