

# UltraScale Architecture Libraries Guide

UG974 (v2016.1) September 15, 2016



## Introduction

---

### Overview

This HDL guide is part of the Vivado® Design Suite documentation collection.

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 UltraScale™ architecture-based devices including the UltraScale and UltraScale+ families, and includes examples of instantiation code for each element. Instantiation templates are also supplied in a separate ZIP file, which you can find on [www.xilinx.com](http://www.xilinx.com) linked to this file or within the Language Templates in the Vivado Design Suite.

Design elements are divided into the following main categories:

**Primitives:** Xilinx components that are native to the architecture you are targeting.

---

## 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 use, implementation, or 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.
- **IP and IP Integrator Catalog:** This component can be instantiated from the IP Catalog. The IP Catalog maintains a library of IP Cores assembled from multiple primitives to form more complex functions, as well as interfaces to help in instantiation of the more complex primitives. References here to the IP Catalog generally refer to the latter, where you use the IP catalog to assist in the use and integration of certain primitives into your design.

# Xilinx Parameterized Macros

---

## About Xilinx Parameterized Macros

This section describes Xilinx Parameterized Macros that can be used with UltraScale™ architecture-based devices. The macros are organized alphabetically.

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

- Name, description, macro group, macro subgroup, and family
  - Schematic symbol
  - Introduction
  - Logic diagram (if any)
  - Port descriptions
  - Design Entry Method
  - Available attributes
  - Example instantiation code
  - Links to additional information
- 

## Enabling Xilinx Parameterized Macros

For XPMs to be recognized by the Vivado Design Suite, they must first be enabled using the XPM\_LIBRARIES property on the project. This is a one-time setup on a project-by project basis. One or more of the available XPM types can be enabled.

In this example, both XPM types are enabled:

```
set_property XPM_LIBRARIES {XPM_CDC XPM_MEMORY} [current_project]
```

In this example, one XPM type is enabled:

```
set_property XPM_LIBRARIES XPM_MEMORY [current_project]
```

After this property is set, you can use any XPM of the type enabled in your design. The property is preserved when the project is closed and reopened.

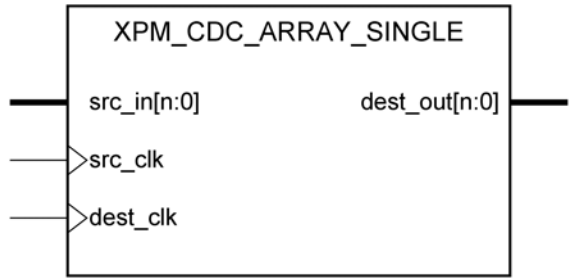
## List of Xilinx Parameterized Macros

Design Element	Description	Macro Subgroup
<a href="#">XPM_CDC_ARRAY_SINGLE</a>	Parameterized Macro: Single-bit Array Synchronizer	CDC
<a href="#">XPM_CDC_ASYNC_RST</a>	Parameterized Macro: Asynchronous Reset Synchronizer	CDC
<a href="#">XPM_CDC_GRAY</a>	Parameterized Macro: Synchronizer via Gray Encoding	CDC
<a href="#">XPM_CDC_HANDSHAKE</a>	Parameterized Macro: Bus Synchronizer with Full Handshake	CDC
<a href="#">XPM_CDC_PULSE</a>	Parameterized Macro: Pulse Transfer	CDC
<a href="#">XPM_CDC_SINGLE</a>	Parameterized Macro: Single-bit Synchronizer	CDC
<a href="#">XPM_CDC_SYNC_RST</a>	Parameterized Macro: Synchronous Reset Synchronizer	CDC
<a href="#">XPM_MEMORY_DPDISTRAM</a>	Parameterized Macro: Dual Port Distributed RAM	Memory
<a href="#">XPM_MEMORY_DPROM</a>	Parameterized Macro: Dual Port ROM	Memory
<a href="#">XPM_MEMORY_SDPROM</a>	Parameterized Macro: Simple Dual Port ROM	Memory
<a href="#">XPM_MEMORY_SDRAM</a>	Parameterized Macro: Simple Dual Port RAM	Memory
<a href="#">XPM_MEMORY_SPRAM</a>	Parameterized Macro: Single Port RAM	Memory
<a href="#">XPM_MEMORY_SPROM</a>	Parameterized Macro: Single Port ROM	Memory
<a href="#">XPM_MEMORY_TDPROM</a>	Parameterized Macro: True Dual Port ROM	Memory
<a href="#">XPM_MEMORY_TDRAM</a>	Parameterized Macro: True Dual Port RAM	Memory

# XPM\_CDC\_ARRAY\_SINGLE

## Parameterized Macro: Single-bit Array Synchronizer

MACRO\_GROUP: XPM  
 MACRO\_SUBGROUP: CDC  
 Families: UltraScale, UltraScale+



X15897-031116

### Introduction

This macro synchronizes an array of single-bit signals from the source clock domain to the destination clock domain. You can define the number of register stages used in the synchronizers. An optional input register can be used to register the input in the source clock domain prior to it being synchronized. You can also enable a simulation feature to generate messages to report any potential misuse of the macro.

**NOTE:** This macro expects that the each bit of the source array is independent, and does not have a defined relationship that needs to be preserved. If each bit of the array has a relationship that needs to be preserved, use the XPM\_CDC\_HANDSHAKE or XPM\_CDC\_GRAY macros.

### Port Description

Port	Direction	Width	Domain	Sense	Handling if Unused	Function
src_clk	Input	1	NA	Rising-edge	Tie to 0	Input clock signal for src_in when SRC_INPUT_REG = 1. Unused when SRC_INPUT_REG = 0.
src_in	Input	WIDTH	src_clk	NA	Required	Input single-bit array to be synchronized to destination clock domain. It is assumed that each bit of the array is unrelated to the others. This is reflected in the constraints applied to this macro.  To transfer a binary value losslessly across the two clock domains, use the XPM_CDC_GRAY macro instead.
dest_clk	Output	1	NA	Rising-edge	Required	Clock signal for the destination clock domain.
dest_out	Output	WIDTH	dest_clk	NA	Required	src_in synchronized to the destination clock domain. This output is registered.

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
DEST_SYNC_FF	Integer	2-10	4	Number of register stages used to synchronize signal in the destination clock domain.
SIM_ASSERT_CHECK	Integer	0-1	0	0: Disable simulation message reporting. Messages related to potential misuse will not be reported. 1: Enable simulation message reporting. Messages related to potential misuse will be reported.
SRC_INPUT_REG	Integer	0-1	1	0: Do not register input (src_in). 1: Register input (src_in) once using src_clk.
WIDTH	Integer	2-1024	2	Width of single-bit array (src_in) that will be synchronized to destination clock domain.

## VHDL Instantiation Template

```
-- Copy the following two statements and paste them before the
-- entity declaration, unless they already exist.

Library xpm;
use xpm.vcomponents.all;

-- <--Cut the following instance declaration and paste it into
-- the architecture statement part of the design-->

-- xpm_cdc_array_single: Clock Domain Crossing Single-bit Array Synchronizer
-- Xilinx Parameterized Macro, Version 2016.1
xpm_cdc_array_single_inst: xpm_cdc_array_single
  generic map (
    -- Common module generics
    DEST_SYNC_FF => 4, -- integer; range: 2-10
    SIM_ASSERT_CHK => 0, -- integer; 0=disable simulation messages, 1=enable simulation messages
    SRC_INPUT_REG => 1, -- integer; 0=do not register input, 1=register input
    WIDTH        => 2  -- integer; range: 2-1024
  )
  port map (
    src_clk => src_clk, -- optional; required when SRC_INPUT_REG = 1
    src_in  => src_in,
    dest_clk => dest_clk,
    dest_out => dest_out
  );
```

```
-- End of xpm_cdc_array_single_inst instance declaration
```

## Verilog Instantiation Template

```
// <--Cut the following instance declaration and paste it into  
// the architecture statement part of the design-->  
  
// xpm_cdc_array_single: Clock Domain Crossing Single-bit Array Synchronizer  
// Xilinx Parameterized Macro, Version 2016.1  
xpm_cdc_array_single #(  
  
    //Common module parameters  
    .DEST_SYNC_FF    (4), // integer; range: 2-10  
    .SIM_ASSERT_CHK  (0), // integer; 0=disable simulation messages, 1=enable simulation messages  
    .SRC_INPUT_REG   (1), // integer; 0=do not register input, 1=register input  
    .WIDTH           (2) // integer; range: 2-1024  
  
) xpm_cdc_array_single_inst (  
  
    .src_clk  (src_clk), // optional; required when SRC_INPUT_REG = 1  
    .src_in   (src_in),  
    .dest_clk (dest_clk),  
    .dest_out (dest_out)  
  
);  
  
// End of xpm_cdc_array_single_inst instance declaration
```

## For More Information (individual items)

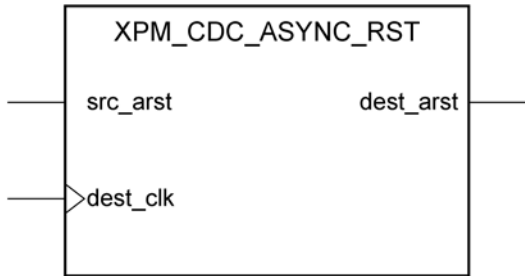
See the [UltraScale Programmable Devices Documentation](#).



# XPM\_CDC\_ASYNC\_RST

## Parameterized Macro: Asynchronous Reset Synchronizer

MACRO\_GROUP: XPM  
 MACRO\_SUBGROUP: CDC  
 Families: UltraScale, UltraScale+



X15902-031116

## Introduction

This macro synchronizes an asynchronous reset signal to the destination clock domain. The resulting reset output will be guaranteed to assert asynchronously in relation to the input but the deassertion of the output will always be synchronous to the destination clock domain. You can define the polarity of the reset signal and the minimal output pulse width of the macro when asserted. The latter is controlled by defining the number of register stages used in the synchronizers.

**NOTE:** The minimum input pulse assertion is dependent on the setup and hold requirement of the reset or set pin of the registers. See the respective DC and AC switching characteristics data sheets for the targeted architecture.

## Port Description

Port	Direction	Width	Domain	Sense	Handling if Unused	Function
src_arst	Input	1	NA	NA	Required	Source asynchronous reset signal.
dest_clk	Input	1	NA	Rising-edge	Required	Clock signal for the destination clock domain.
dest_arst	Output	1	dest_clk	NA	Required	The src_arst asynchronous reset signal synchronized to destination clock domain. This output is registered.  <b>NOTE:</b> The signal asserts asynchronously but deasserts synchronously to dest_clk. The width of the reset signal is at least (DEST_SYNC_FF*dest_clk) period.

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
DEST_SYNC_FF	Integer	2-10	4	Number of register stages used to synchronize signal in the destination clock domain. This parameter also determines the minimum width of the asserted reset signal.
RST_ACTIVE_HIGH	Integer	0-1	0	Defines the polarity of the asynchronous reset signal. <ul style="list-style-type: none"> <li>0: Active low asynchronous reset signal</li> <li>1: Active high asynchronous reset signal</li> </ul>

## VHDL Instantiation Template

```
-- Copy the following two statements and paste them before the
-- entity declaration, unless they already exist.

Library xpm;
use xpm.vcomponents.all;

-- <--Cut the following instance declaration and paste it into
-- the architecture statement part of the design-->

-- xpm_cdc_async_rst: Clock Domain Crossing Asynchronous Reset Synchronizer
-- Xilinx Parameterized Macro, Version 2016.1
xpm_cdc_async_rst_inst: xpm_cdc_async_rst
    generic map (
        -- Common module parameters
        DEST_SYNC_FF => 4, -- integer; range: 2-10
        RST_ACTIVE_HIGH => 0 -- integer; 0=active low reset, 1=active high reset
    )
    port map (
        src_arst => src_arst,
        dest_clk => dest_clk,
        dest_arst => dest_arst
    );

-- End of xpm_cdc_async_rst_inst instance declaration
```

## Verilog Instantiation Template

```
// <--Cut the following instance declaration and paste it into
// the architecture statement part of the design-->

// xpm_cdc_async_rst: Clock Domain Crossing Asynchronous Reset Synchronizer
// Xilinx Parameterized Macro, Version 2016.1
xpm_cdc_async_rst #(

    //Common module parameters
    .DEST_SYNC_FF      (4), // integer; range: 2-10
    .RST_ACTIVE_HIGH (0) // integer; 0=active low reset, 1=active high reset

) xpm_cdc_async_rst_inst (

    .src_arst  (src_arst),
    .dest_clk  (dest_clk),
    .dest_arst (dest_arst)

);

// End of xpm_cdc_async_rst_inst instance declaration
```

## For More Information (individual items)

See the [UltraScale Programmable Devices Documentation](#).

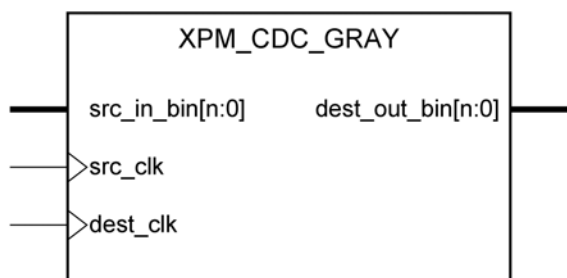
## XPM\_CDC\_GRAY

### Parameterized Macro: Synchronizer via Gray Encoding

MACRO\_GROUP: XPM

MACRO\_SUBGROUP: CDC

Families: UltraScale, UltraScale+



X15898-031116

## Introduction

This macro synchronizes a binary input from the source clock domain to the destination clock domain using Gray code. It takes the input binary signal, translates it into Gray code and registers it, synchronizes it to the destination clock domain, and then translates it back to a binary signal. You can define the number of register stages used in the synchronizers. You can also enable a simulation feature to generate messages to report any potential misuse of the macro.

Since this macro uses Gray encoding, the binary value provided to the macro must only increment or decrement by one to ensure that the signal being synchronized has two successive values that only differ by one bit. This will ensure lossless synchronization of a Gray coded bus. If the behavior of the binary value is not compatible to Gray encoding, use the XPM\_CDC\_HANDSHAKE macro or an alternate method of synchronizing the data to the destination clock domain.

An additional option (SIM\_LOSSLESS\_GRAY\_CHK) is provided to report an error message when any binary input values are found to violate the Gray coding rule where two successive values must only increment or decrement by one.

**NOTE:** When the XPM\_CDC\_GRAY module is used in a design and report\_cdc is run, the synchronizer in this module is reported as a warning of type CDC-6, Multi-bit synchronized with ASYNC\_REG property. This warning is safe to ignore since the bus that is synchronized is gray-coded.

## Port Description

Port	Direction	Width	Domain	Sense	Handling if Unused	Function
src_clk	Input	1	NA	Rising-edge	Required	Source clock.
src_in_bin	Input	WIDTH	src_clk	NA	Required	Binary input bus that will be synchronized to the destination clock domain.
dest_clk	Input	1	NA	Rising-edge	Required	Destination clock.
dest_out_bin	Output	WIDTH	dest_clk	NA	Required	Binary input bus (src_in_bin) synchronized to destination clock domain. This output is combinatorial.

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
DEST_SYNC_FF	Integer	2-10	4	Number of register stages used to synchronize signal in the destination clock domain.
SIM_ASSERT_CHECK	Integer	0-1	0	0: Disable simulation message reporting, messages related to potential misuse will not be reported. 1: Enable simulation message reporting, messages related to potential misuse will be reported.
SIM_LOSSLESS_GRAY_CHK	Integer	0-1	0	0: Disable simulation message that reports whether src_in_bin is incrementing or decrementing by one, guaranteeing lossless synchronization of a gray coded bus. 1: Enable simulation message that reports whether src_in_bin is incrementing or decrementing by one, guaranteeing lossless synchronization of a gray coded bus.
WIDTH	Integer	2-32	2	Width of binary input bus that will be synchronized to destination clock domain.

## VHDL Instantiation Template

```
-- Copy the following two statements and paste them before the
-- entity declaration, unless they already exist.

Library xpm;
use xpm.vcomponents.all;

-- <--Cut the following instance declaration and paste it into
-- the architecture statement part of the design-->

-- xpm_cdc_gray: Clock Domain Crossing Synchronizer via Gray Encoding
-- Xilinx Parameterized Macro, Version 2016.1
xpm_cdc_gray_inst: xpm_cdc_gray
  generic map (
    -- Common module generics
    DEST_SYNC_FF          => 4, -- integer; range: 2-10
    SIM_ASSERT_CHK        => 0, -- integer; 0=disable simulation messages, 1=enable simulation messages
    SIM_LOSSLESS_GRAY_CHK => 0, -- integer; 0=disable lossless check, 1=enable lossless check
    WIDTH                 => 2  -- integer; range: 2-32
  )
  port map (
    src_clk      => src_clk,
    src_in_bin   => src_in_bin,
    dest_clk     => dest_clk,
    dest_out_bin => dest_out_bin
  );

-- End of xpm_cdc_gray_inst instance declaration
```

## Verilog Instantiation Template

```
// <--Cut the following instance declaration and paste it into
// the architecture statement part of the design-->

// xpm_cdc_gray: Clock Domain Crossing Synchronizer via Gray Encoding
// Xilinx Parameterized Macro, Version 2016.1
xpm_cdc_gray #(
  //Common module parameters
  .DEST_SYNC_FF (4), // integer; range: 2-10
  .SIM_ASSERT_CHK (0), // integer; 0=disable simulation messages, 1=enable simulation messages
  .SIM_LOSSLESS_GRAY_CHK (0), // integer; 0=disable lossless check, 1=enable lossless check
  .WIDTH (2) // integer; range: 2-32
) xpm_cdc_gray_inst (
  .src_clk (src_clk),
  .src_in_bin (src_in_bin),
  .dest_clk (dest_clk),
  .dest_out_bin (dest_out_bin)
);

// End of xpm_cdc_gray_inst instance declaration
```

## For More Information (individual items)

See the [UltraScale Programmable Devices Documentation](#).

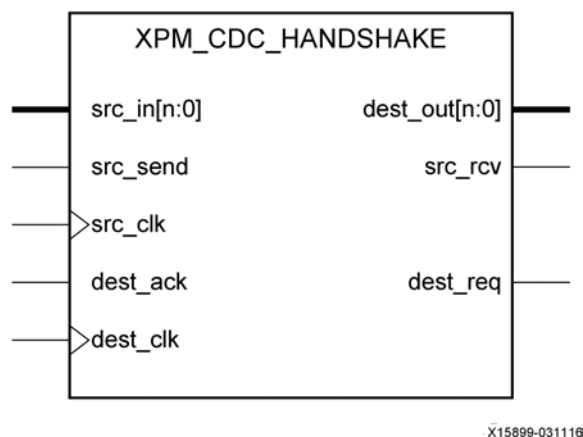
# XPM\_CDC\_HANDSHAKE

## Parameterized Macro: Bus Synchronizer with Full Handshake

MACRO\_GROUP: XPM

MACRO\_SUBGROUP: CDC

Families: UltraScale, UltraScale+



## Introduction

This macro uses a handshake signaling to transfer an input bus from the source clock domain to the destination clock domain. One example of when this macro should be used is when the data that is being transferred is not compatible with the XPM\_CDC\_GRAY macro that uses Gray encoding. For this macro to function correctly, a full handshake - an acknowledgement that the data transfer was received and a resetting of the handshake signals - must be completed before another data transfer is initiated.

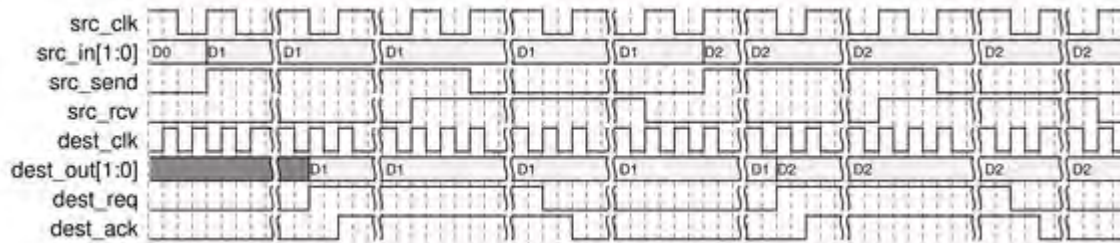
You can define the number of register stages used in the synchronizers to transfer the handshake signals between the clock domains individually. You can also include internal handshake logic to acknowledge the receipt of data on the destination clock domain. When this feature is enabled, the output (`dest_out`) must be consumed immediately when the data valid (`dest_req`) is asserted.

You can also enable a simulation feature to generate messages to report any potential misuse of the macro. These messages will generate errors when the signaling provided to the macro violates the usage guidance above.

**NOTE:** When the XPM\_CDC\_HANDSHAKE module is used in a design and `report_cdc` is run, the data bus that is synchronized in this module is reported as a warning of type CDC-15, Clock Enable Controlled CDC. This warning is safe to ignore.

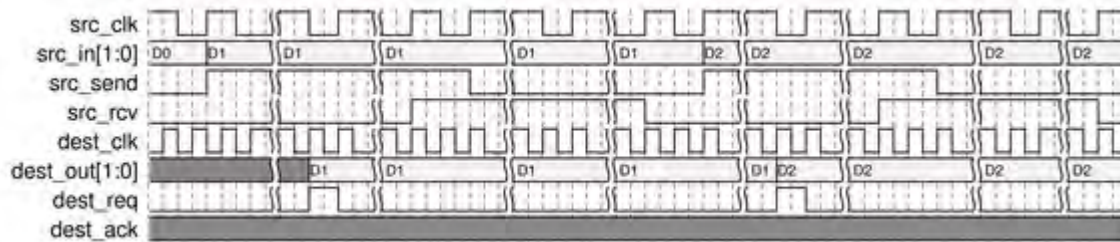
## External Handshake

The following waveform shows how back-to-back data is sent when the external handshake option is used.



## Internal Handshake

The following waveform shows how back-to-back data is sent when the internal handshake option is enabled.



## Port Description

Port	Direction	Width	Domain	Sense	Handling if Unused	Function
src_clk	Input	1	NA	Rising-edge	Required	Source clock.
src_in	Input	WIDTH	src_clk	NA	Required	Input bus that will be synchronized to the destination clock domain.
src_send	Input	1	src_clk	Active-high	Required	<p>Assertion of this signal allows the src_in bus to be synchronized to the destination clock domain.</p> <ul style="list-style-type: none"> <li>This signal should only be asserted when src_rcv is deasserted, indicating that the previous data transfer is complete.</li> <li>This signal should only be deasserted once src_rcv is asserted, acknowledging that the src_in has been received by the destination logic.</li> </ul>



Port	Direction	Width	Domain	Sense	Handling if Unused	Function
src_rcv	Output	1	src_clk	Active-high	Required	Acknowledgement from destination logic that src_in has been received. This signal will be deasserted once destination handshake has fully completed, thus completing a full data transfer. This output is registered.
dest_clk	Input	1	NA	Rising-edge	Required	Destination clock.
dest_out	Output	WIDTH	dest_clk	NA	Required	Input bus (src_in) synchronized to destination clock domain. This output is registered.
dest_req	Output	1	dest_clk	Active-high	Required	Assertion of this signal indicates that new dest_out data has been received and is ready to be used or captured by the destination logic. <ul style="list-style-type: none"> <li>When DEST_EXT_HSK = 1, this signal will deassert once the source handshake acknowledges that the destination clock domain has received the transferred data.</li> <li>When DEST_EXT_HSK = 0, this signal asserts for one clock period when dest_out bus is valid.</li> </ul> This output is registered.
dest_ack	Input	1	dest_clk	Active-high	Tie to 0	Destination logic acknowledgement if DEST_EXT_HSK = 1. Unused when DEST_EXT_HSK = 0. Asserting this signal indicates that data on dest_out has been captured by the destination logic. This signal should be deasserted once dest_req is deasserted, completing the handshake on the destination clock domain and indicating that the destination logic is ready for a new data transfer.

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
DEST_EXT_HSK	Integer	0-1	1	0: An internal handshake will be implemented in the module to acknowledge receipt of data on the destination clock domain. When using this option, the valid dest_out output must be consumed immediately to avoid any data loss.  1: You must implement external handshake logic to acknowledge receipt of data on the destination clock domain.
DEST_SYNC_FF	Integer	2-10	4	Number of register stages used to synchronize signal in the destination clock domain.
SIM_ASSERT_CHECK	Integer	0-1	0	0: Disable simulation message reporting, messages related to potential misuse will not be reported.  1: Enable simulation message reporting, messages related to potential misuse will be reported.
SRC_SYNC_FF	Integer	2-10	4	Number of register stages used to synchronize signal in the source clock domain.
WIDTH	Integer	1-1024	1	Width of bus that will be synchronized to destination clock domain.

## VHDL Instantiation Template

```
-- Copy the following two statements and paste them before the
-- entity declaration, unless they already exist.

Library xpm;
use xpm.vcomponents.all;

-- <--Cut the following instance declaration and paste it into
-- the architecture statement part of the design-->

-- xpm_cdc_handshake: Clock Domain Crossing Bus Synchronizer with Full Handshake
-- Xilinx Parameterized Macro, Version 2016.1
xpm_cdc_handshake_inst: xpm_cdc_handshake
    generic map (
        -- Common module generics
        DEST_EXT_HSK => 1, -- integer; 0=user handshake, 1=internal handshake
        DEST_SYNC_FF => 4, -- integer; range: 2-10
        SIM_ASSERT_CHK => 0, -- integer; 0=disable simulation messages, 1=enable simulation messages
        SRC_SYNC_FF => 4, -- integer; range: 2-10
        WIDTH => 1 -- integer; range: 1-1024
    )
    port map (
        src_clk => src_clk,
        src_in => src_in,
        src_send => src_send,
        src_rcv => src_rcv,
```

```

    dest_clk => dest_clk,
    dest_req => dest_req,
    dest_ack => dest_ack, -- optional; required when DEST_EXT_HSK = 1
    dest_out => dest_out
);

-- End of xpm_cdc_handshake_inst instance declaration

```

## Verilog Instantiation Template

```

// <--Cut the following instance declaration and paste it into
// the architecture statement part of the design-->

// xpm_cdc_handshake: Clock Domain Crossing Synchronizer with Full Handshake
// Xilinx Parameterized Macro, Version 2016.1
xpm_cdc_handshake #(

    //Common module parameters
    .DEST_EXT_HSK    (1), // integer; 0=user handshake, 1=internal handshake
    .DEST_SYNC_FF   (4), // integer; range: 2-10
    .SIM_ASSERT_CHK (0), // integer; 0=disable simulation messages, 1=enable simulation messages
    .SRC_SYNC_FF    (4), // integer; range: 2-10
    .WIDTH          (1) // integer; range: 1-1024

) xpm_cdc_handshake_inst (

    .src_clk  (src_clk),
    .src_in   (src_in),
    .src_send (src_send),
    .src_rcv  (src_rcv),
    .dest_clk (dest_clk),
    .dest_req (dest_req),
    .dest_ack (dest_ack), // optional; required when DEST_EXT_HSK = 1
    .dest_out (dest_out)
);

// End of xpm_cdc_handshake_inst instance declaration

```

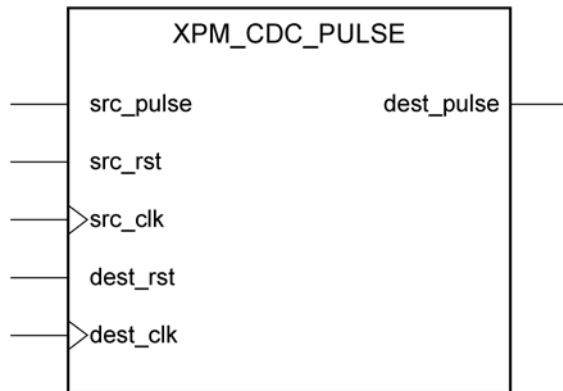
## For More Information (individual items)

See the [UltraScale Programmable Devices Documentation](#).

# XPM\_CDC\_PULSE

## Parameterized Macro: Pulse Transfer

MACRO\_GROUP: XPM  
 MACRO\_SUBGROUP: CDC  
 Families: UltraScale, UltraScale+



X15900-031116

## Introduction

This macro synchronizes a pulse in the source clock domain to the destination clock domain. A pulse of any size in the source clock domain, if initiated correctly, will generate a pulse the size of a single destination clock period. You can define the number of register stages used in the synchronizers. An optional source and destination reset may be used to reset the pulse transfer logic. You can also enable a simulation feature to generate messages which report any potential misuse of the macro.

The implementation of this macro requires some feedback logic. When simulating the macro without the optional reset signals, the input pulse signal (src\_pulse) must always be defined since there is no reset logic to recover from an undefined or 'x' propagating through the macro.

This macro also requires the following minimum gap between subsequent pulse inputs:

$$2 * (\text{larger}(\text{src\_clk\_period}, \text{dest\_clk\_period}))$$

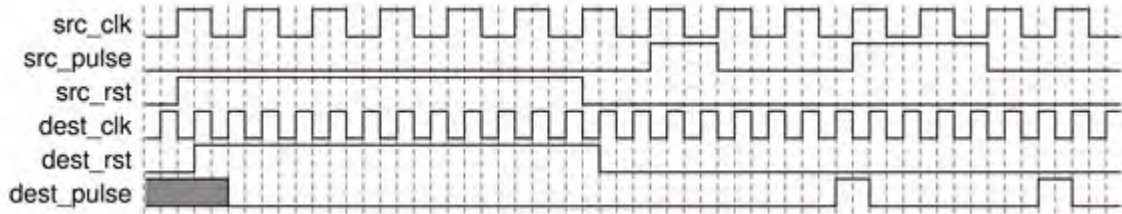
The minimum gap is measured between the falling edge of a src\_pulse to the rising edge of the next src\_pulse. This minimum gap will guarantee that each rising edge of src\_pulse will generate a pulse the size of one dest\_clk period in the destination clock domain.

When using the optional reset signals, src\_rst and dest\_rst must be asserted simultaneously for at least the following duration to fully reset all the logic in the macro:

$$((\text{DEST\_SYNC\_FF}+2) * \text{dest\_clk\_period}) + (2 * \text{src\_clk\_period})$$

When reset is asserted, the input pulse signal should not toggle and the output pulse signal is not valid and should be ignored.

The following waveform demonstrates how to reset the macro and transfer back-to-back pulses while abiding the minimum gap between each pulse.



## Port Description

Port	Direction	Width	Domain	Sense	Handling if Unused	Function
src_clk	Input	1	NA	Rising-edge	Required	Source clock.
src_rst	Input	1	src_clk	Active-high	Tie to 0	Resets all logic in the source clock domain. <ul style="list-style-type: none"> <li>Unused when RST_USED = 0</li> <li>Source reset signal if RST_USED = 1</li> </ul> To fully reset the macro, src_rst and dest_rst must be asserted simultaneously for at least $(DEST\_SYNC\_FF+2)*dest\_clk\_period + (2*src\_clk\_period)$
src_pulse	Input	1	src_clk	Rising-edge	Required	The rising edge of this signal initiates a pulse transfer to the destination clock domain. <p>The minimum gap between each pulse transfer must be at least <math>2*(larger(src\_clk\ period, dest\_clk\ period))</math>. This is measured between the falling edge of a src_pulse to the rising edge of the next src_pulse. This minimum gap will guarantee that each rising edge of src_pulse will generate a pulse the size of one dest_clk period in the destination clock domain.</p> When RST_USED = 1, pulse transfers will not be guaranteed while src_rst and/or dest_rst are asserted.
dest_clk	Input	1	NA	Rising-edge	Required	Destination clock.

Port	Direction	Width	Domain	Sense	Handling if Unused	Function
dest_rst	Input	1	dest_clk	Active-high	Tie to 0	Resets all logic in destination clock domain. <ul style="list-style-type: none"> <li>Unused when RST_USED = 0</li> <li>Destination reset signal if RST_USED = 1</li> </ul> To fully reset the macro, src_rst and dest_rst must be asserted simultaneously for at least $((\text{DEST\_SYNC\_FF}+2)*\text{dest\_clk\_period}) + (2*\text{src\_clk\_period})$ .
dest_pulse	Output	1	dest_clk	Active-high	Required	Outputs a pulse the size of one dest_clk period when a pulse transfer is correctly initiated on src_pulse input. This output is combinatorial.

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
DEST_SYNC_FF	Integer	2–10	4	Number of register stages used to synchronize signal in the destination clock domain.
RST_USED	Integer	0-1	1	0 : No resets implemented. 1 : Resets implemented. When RST_USED = 0, src_pulse input must always be defined during simulation since there is no reset logic to recover from an x-propagating through the module.
SIM_ASSERT_CHECK	Integer	0-1	0	0: Disable simulation message reporting, messages related to potential misuse will not be reported. 1: Enable simulation message reporting, messages related to potential misuse will be reported.

## VHDL Instantiation Template

```
-- Copy the following two statements and paste them before the
-- entity declaration, unless they already exist.

Library xpm;
use xpm.vcomponents.all;

-- <--Cut the following instance declaration and paste it into
-- the architecture statement part of the design-->

-- xpm_cdc_pulse: Clock Domain Crossing Pulse Transfer
-- Xilinx Parameterized Macro, Version 2016.1
xpm_cdc_pulse_inst: xpm_cdc_pulse
  generic map (
    -- Common module generics
    DEST_SYNC_FF    => 4, -- integer; range: 2-10
    RST_USED        => 1, -- integer; 0=no reset, 1=implement reset
    SIM_ASSERT_CHK  => 0  -- integer; 0=disable simulation messages, 1=enable simulation messages
  )
  port map (
    src_clk    => src_clk,
    src_rst    => src_rst,  -- optional; required when RST_USED = 1
    src_pulse  => src_pulse,
    dest_clk   => dest_clk,
    dest_rst   => dest_rst, -- optional; required when RST_USED = 1
    dest_pulse => dest_pulse
  );

-- End of xpm_cdc_pulse_inst instance declaration
```

## Verilog Instantiation Template

```
// <--Cut the following instance declaration and paste it into
// the architecture statement part of the design-->

// xpm_cdc_pulse: Clock Domain Crossing Pulse Transfer
// Xilinx Parameterized Macro, Version 2016.1
xpm_cdc_pulse #(
  //Common module parameters
  .DEST_SYNC_FF  (4), // integer; range: 2-10
  .RST_USED      (1), // integer; 0=no reset, 1=implement reset
  .SIM_ASSERT_CHK (0) // integer; 0=disable simulation messages, 1=enable simulation messages
) xpm_cdc_pulse_inst (
  .src_clk    (src_clk),
  .src_rst    (src_rst),  // optional; required when RST_USED = 1
  .src_pulse  (src_pulse),
  .dest_clk   (dest_clk),
  .dest_rst   (dest_rst), // optional; required when RST_USED = 1
  .dest_pulse (dest_pulse)
);

// End of xpm_cdc_pulse_inst instance declaration
```

## For More Information (individual items)

See the [UltraScale Programmable Devices Documentation](#).

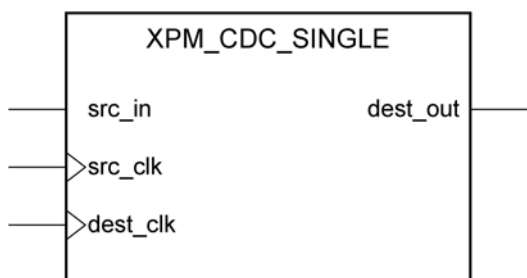
## XPM\_CDC\_SINGLE

### Parameterized Macro: Single-bit Synchronizer

MACRO\_GROUP: XPM

MACRO\_SUBGROUP: CDC

Families: UltraScale, UltraScale+



X15896-031116

## Introduction

This macro synchronizes a one bit signal from the source clock domain to the destination clock domain. You can define the number of register stages used in the synchronizers. An optional input register may be used to register the input in the source clock domain prior to it being synchronized. You can also enable a simulation feature to generate messages to report any potential misuse of the macro.

## Port Description

Port	Direction	Width	Domain	Sense	Handling if Unused	Function
src_clk	Input	1	NA	Rising-edge	Tie to 0	Input clock signal for src_in if SRC_INPUT_REG = 1. Unused when SRC_INPUT_REG = 0.
src_in	Input	1	src_clk	NA	Required	Input signal to be synchronized to dest_clk domain.
dest_clk	Input	1	NA	Rising-edge	Required	Clock signal for the destination clock domain.
dest_out	Output	1	dest_clk	NA	Required	src_in synchronized to the destination clock domain. This output is registered.

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	No



## Available Attributes

Attribute	Type	Allowed Values	Default	Description
DEST_SYNC_FF	Integer	2-10	4	Number of register stages used to synchronize signal in the destination clock domain.
SIM_ASSERT_CHECK	Integer	0-1	0	0: Disable simulation message reporting, messages related to potential misuse will not be reported. 1: Enable simulation message reporting, messages related to potential misuse will be reported.
SRC_INPUT_REG	Integer	0-1	1	0: Do not register input (src_in) 1: Register input (src_in) once using src_clk

## VHDL Instantiation Template

```
-- Copy the following two statements and paste them before the
-- entity declaration, unless they already exist.

Library xpm;
use xpm.vcomponents.all;

-- <--Cut the following instance declaration and paste it into
-- the architecture statement part of the design-->

-- xpm_cdc_single: Clock Domain Crossing Single-bit Synchronizer
-- Xilinx Parameterized Macro, Version 2016.1
xpm_cdc_single_inst: xpm_cdc_single
    generic map (
        DEST_SYNC_FF => 4, -- integer; range: 2-10
        SIM_ASSERT_CHK => 0, -- integer; 0=disable simulation messages, 1=enable simulation messages
        SRC_INPUT_REG => 1 -- integer; 0=do not register input, 1=register input
    )
    port map (
        src_clk => src_clk, -- optional; required when SRC_INPUT_REG = 1
        src_in => src_in,
        dest_clk => dest_clk,
        dest_out => dest_out
    );

-- End of xpm_cdc_single_inst instance declaration
```

## Verilog Instantiation Template

```
// <--Cut the following instance declaration and paste it into
// the architecture statement part of the design-->

// xpm_cdc_single: Clock Domain Crossing Single-bit Synchronizer
// Xilinx Parameterized Macro, Version 2016.1
xpm_cdc_single #(

    //Common module parameters
    .DEST_SYNC_FF (4), // integer; range: 2-10
    .SIM_ASSERT_CHK (0), // integer; 0=disable simulation messages, 1=enable simulation messages
    .SRC_INPUT_REG (1) // integer; 0=do not register input, 1=register input

) xpm_cdc_single_inst (
```

```
.src_clk (src_clk), // optional; required when SRC_INPUT_REG = 1  
.src_in  (src_in),  
.dest_clk (dest_clk),  
.dest_out (dest_out)  
  
);  
  
// End of xpm_cdc_single_inst instance declaration
```

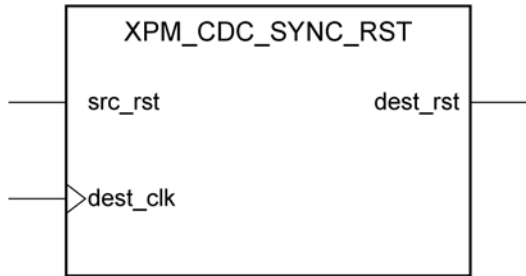
## For More Information

See the [UltraScale Programmable Devices Documentation](#).

# XPM\_CDC\_SYNC\_RST

## Parameterized Macro: Synchronous Reset Synchronizer

MACRO\_GROUP: XPM  
 MACRO\_SUBGROUP: CDC  
 Families: UltraScale, UltraScale+



X15901-031116

## Introduction

This macro synchronizes a reset signal to the destination clock domain. Unlike the XPM\_CDC\_ASYNC\_RST macro, the generated output will both assert and deassert synchronously to the destination clock domain. You can define the number of register stages used in the synchronizers and the initial value of these registers after configuration. An optional input register may be used to register the input in the source clock domain prior to it being synchronized. You can also enable a simulation feature to generate messages which report any potential misuse of the macro.

## Port Description

Port	Direction	Width	Domain	Sense	Handling if Unused	Function
src_rst	Input	1	NA	NA	Required	Source reset signal.
dest_clk	Input	1	NA	Rising-edge	Required	Clock signal for the destination clock domain.
dest_rst	Output	1	dest_clk	NA	Required	src_rst synchronized to the destination clock domain. This output is registered.

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
DEST_SYNC_FF	Integer	2-10	4	Number of register stages used to synchronize the signal in the destination clock domain.
INIT	Integer	0-1	1	0: Initializes synchronization registers to 0. 1: Initializes synchronization registers to 1. Because the synchronization registers are initialized, there is no complete x-propagation behavior modeled in this macro. For complete x-propagation modelling, use the XPM_CDC_SINGLE macro.
SIM_ASSERT_CHECK	Integer	0-1	0	0: Disable simulation message reporting, messages related to potential misuse will not be reported. 1: Enable simulation message reporting, messages related to potential misuse will be reported.

## VHDL Instantiation Template

```
-- Copy the following two statements and paste them before the
-- entity declaration, unless they already exist.

Library xpm;
use xpm.vcomponents.all;

-- <--Cut the following instance declaration and paste it into
-- the architecture statement part of the design-->

-- xpm_cdc_sync_rst: Clock Domain Crossing Synchronous Reset Synchronizer
-- Xilinx Parameterized Macro, Version 2016.1
xpm_cdc_sync_rst_inst: xpm_cdc_sync_rst
  generic map (
    -- Common module parameters
    DEST_SYNC_FF => 4, -- integer; range: 2-10
    INIT         => 1, -- integer; 0=initialize synchronization registers to 0,
                --           1=initialize synchronization registers to 1
    SIM_ASSERT_CHK => 0 -- integer; 0=disable simulation messages, 1=enable simulation messages
  )
  port map (
    src_rst => src_rst,
    dest_clk => dest_clk,
    dest_rst => dest_rst
  );

-- End of xpm_cdc_sync_rst_inst instance declaration
```

## Verilog Instantiation Template

```
// <--Cut the following instance declaration and paste it into
// the architecture statement part of the design-->

// xpm_cdc_sync_rst: Clock Domain Crossing Synchronous Reset Synchronizer
// Xilinx Parameterized Macro, Version 2016.1
xpm_cdc_sync_rst #(

    //Common module parameters
    .DEST_SYNC_FF    (4), // integer; range: 2-10
    .INIT            (1), // integer; 0=initialize synchronization registers to 0,
                        //           1=initialize synchronization registers to 1
    .SIM_ASSERT_CHK (0) // integer; 0=disable simulation messages, 1=enable simulation messages

) xpm_cdc_sync_rst_inst (

    .src_rst  (src_rst),
    .dest_clk (dest_clk),
    .dest_rst (dest_rst)

);

// End of xpm_cdc_sync_rst_inst instance declaration
```

## For More Information (individual items)

See the [UltraScale Programmable Devices Documentation](#).

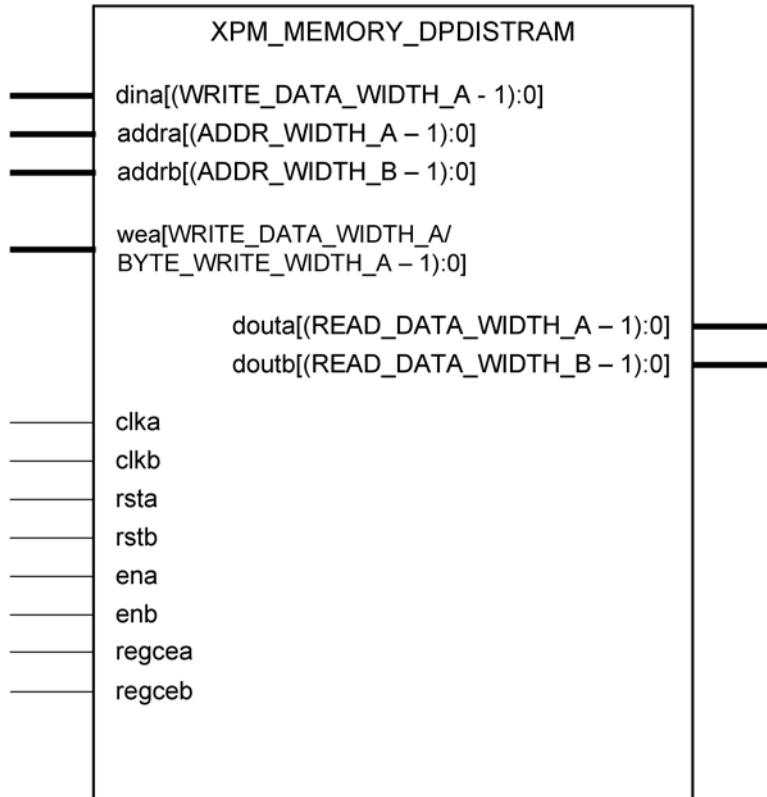
# XPM\_MEMORY\_DPDISTRAM

Parameterized Macro: Dual Port Distributed RAM

MACRO\_GROUP: XPM

MACRO\_SUBGROUP: MEMORY

Families: UltraScale, UltraScale+



X16219-040416

## Introduction

This Macro is used to Instantiate Dual Port Distributed RAM. Port-A can be used to perform both Read and Write operations to the memory and simultaneously Port-B can be used to perform the Read operations from the memory. Write Operations are not allowed through port-B.

The following describes the basic read and write port usage of an XPM\_MEMORY instance. It does not distinguish between ports A and B.

- All synchronous signals are sensitive to the rising edge of clk[a|b], which is assumed to be a buffered and toggling clock signal behaving according to target device and memory primitive requirements.
- A read operation is implicitly performed to address addr[a|b] combinatorially. The data output is registered each clk[a|b] cycle that en[a|b] is asserted.
- Read data appears on the dout[a|b] port READ\_LATENCY\_[A|B] clk[a|b] cycles after the associated read operation.
- A write operation is explicitly performed, writing din[a] to address addr[a], when both en[a] and we[a] are asserted on each clk[a] cycle.
- All read and write operations are gated by the value of en[a|b] on the initiating clk[a|b] cycle, regardless of input or output latencies. The addr[a] and we[a] inputs have no effect when en[a] is de-asserted on the coincident clk[a] cycle.
- For each clk[a|b] cycle that rst[a|b] is asserted, the final output register is immediately but synchronously reset to READ\_RESET\_VALUE\_[A|B], irrespective of READ\_LATENCY\_[A|B].
- For each clk[a|b] cycle that regcea[a|b] is asserted and rst[a|b] is de-asserted, the final output register captures and outputs the value from the previous pipeline register.
- Undriven or unknown values provided on module inputs will produce undefined memory array and output port behavior.

**NOTE:** When the attribute "CLOCKING\_MODE" is set to "common\_clock", then all the read/write operations to the memory through port-A and port-B are performed on clka. If this attribute is set to "independent\_clock", then read/write operations through port-A are performed based on clka, and read/write operations through port-B are performed based on clkb.

## Port Description

Port	Direction	Width	Sense	Handling if unused	Description
clka	Input	1	Rising-edge	Required	Port A Clock
clkb	Input	1	Rising-edge	Required	Port B Clock
rsta	Input	1	NA	Required	Port A final output register reset to READ_RESET_VALUE_A
rstb	Input	1	NA	Required	Port B final output register reset to READ_RESET_VALUE_B
ena	Input	1	NA	Required	Port A Enable
enb	Input	1	NA	Required	Port B Enable
regcea	Input	1	NA	Required	Port A final output pipeline register clock enable
regceb	Input	1	NA	Required	Port B final output pipeline register clock enable
wea	Input	WRITE_DATA_WIDTH_A / BYTE_WRITE_WIDTH_A	NA	Required	Port A Write Enable

Port	Direction	Width	Sense	Handling if unused	Description
addra	Input	ADDR_WIDTH_A	NA	Required	Port A Address
addrb	Input	ADDR_WIDTH_B	NA	Required	Port B Address
dina	Input	WRITE_DATA_WIDTH_A	NA	Required	Port A Write Data
douta	Output	READ_DATA_WIDTH_A	NA	Required	Port A output data
doutb	Output	READ_DATA_WIDTH_B	NA	Required	Port B output data

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
MEMORY_SIZE	Integer	>=1	2048	Total memory array size, in bits
CLOCKING_MODE	string	"common_clock", "independent_clock"	"common_clock"	Clocking Mode
WRITE_DATA_WIDTH_[A]	Integer	>=1	32	Width of the port [A] write data port, din[a]
BYTE_WRITE_WIDTH_[A]	Integer	8, 9, WRITE_DATA_WIDTH_[A]	WRITE_DATA_WIDTH_[A]	Width of each byte, in bits, for enabled byte-wide write functionality of port [A]
READ_DATA_WIDTH_[A B]	Integer	>=1	WRITE_DATA_WIDTH_[A]	Width of the port [A] output data port, dout[a]
READ_RESET_VALUE_[B]	string	All type-legal values (hex string)	"0"	Initial and reset value of final register stage in port [B] output data path
READ_LATENCY_[A B]	Integer	>=1	2	Number of register stages in the port [A B] read data path
MEMORY_INIT_FILE	Unspecified	"none", "<file_name>"	"none"	File name for memory content initialization.



Attribute	Type	Allowed Values	Default	Description
MEMORY_INIT_PARAM	Unspecified	"", valid hex String	""	<p>Initialize the contents of the memory using a parameter.</p> <p>Specify "" or "0" (including quotes) for no memory initialization through a parameter, or specify the string containing the hex characters. Each location must be separated by a delimiter (.).</p> <p>The parameter format must be ASCII and consist of only hexadecimal values organized into the specified depth by the narrowest data width generic value of the memory. For example, if the narrowest data width is 8, and the depth of memory is 8 locations, then the parameter value should be passed as shown below (0xAB in the below string is the 0th location and 0x78 is the 7th location).</p> <pre>MEMORY_INIT_PARAM = "AB,CD,EF,1,2,34,56,78"</pre> <p>Maximum supported memory size when MEMORY_INIT_PARAM is used is 4k bits.</p>
USE_MEMORY_INIT	Integer	0, 1	0	<ul style="list-style-type: none"> <li>1 = Enable the generation of below message</li> <li>0 = Disable the generation of below message</li> </ul> <p><b>NOTE:</b> This message gets generated only when there is no Memory Initialization specified either through file or parameter. INFO: MEMORY_INIT_FILE and MEMORY_INIT_PARAM together specifies no memory initialization. Initial memory contents will be all 0's</p>
WAKEUP_TIME	string	"disable_sleep", "use_sleep_pin"	"disable_sleep"	<p>Number of clk[a] cycles to wake, following de-assertion of sleep. "use_sleep_pin" has wake up time of 2 clock cycles.</p>
MESSAGE_CONTROL	Integer	0, 1	1	<p>Module messaging control</p> <ul style="list-style-type: none"> <li>0 = Disable dynamic messages (assertions)</li> <li>1 = Enable dynamic messages (assertions)</li> </ul>

## VHDL Instantiation Template

```
-- Copy the following two statements and paste them before the
-- entity declaration, unless they already exist.

Library xpm;
use xpm.vcomponents.all;

-- <--Cut the following instance declaration and paste it into
-- the architecture statement part of the design-->

-- xpm_memory_dpdistram: Dual Port Distributed RAM
-- Xilinx Parameterized Macro, Version 2016.1
xpm_memory_dpdistram_inst : xpm_memory_dpdistram
  generic map (
    -- Common module generics
    MEMORY_SIZE      => 2048,          --positive integer
    CLOCKING_MODE    => "common_clock",--string; "common_clock", "independent_clock"
    MEMORY_INIT_FILE => "none",        --string; "none" or "<filename>.mem"
    MEMORY_INIT_PARAM => "",          --string;
    USE_MEM_INIT     => 1,            --integer; 0,1
    MESSAGE_CONTROL  => 0,            --integer; 0,1

    -- Port A module generics
    WRITE_DATA_WIDTH_A => 32,          --positive integer
    READ_DATA_WIDTH_A  => 32,          --positive integer
    BYTE_WRITE_WIDTH_A => 32,          --integer; 8, 9, or WRITE_DATA_WIDTH_A value
    ADDR_WIDTH_A       => 6,          --positive integer
    READ_RESET_VALUE_A => "0",        --string
    READ_LATENCY_A     => 2,          --non-negative integer

    -- Port B module generics
    READ_DATA_WIDTH_B  => 32,          --positive integer
    ADDR_WIDTH_B       => 6,          --positive integer
    READ_RESET_VALUE_B => "0",        --string
    READ_LATENCY_B     => 2,          --non-negative integer
  )
  port map (
    -- Port A module ports
    clka => clka,
    rsta => rsta,
    ena  => ena,
    regcea => '1', --do not change
    wea  => wea,
    addra => addra,
    dina => dina,
    douta => douta,

    -- Port B module ports
    clkb => clkb,
    rstb => rstb,
    enb  => enb,
    regceb => '1', --do not change
    addrb => addrb,
    doutb => doutb
  );

-- End of xpm_memory_dpdistram_inst instance declaration
```

## Verilog Instantiation Template

```
// <--Cut the following instance declaration and paste it into
// the architecture statement part of the design-->

// xpm_memory_dpdistram: Dual Port distributed RAM
// Xilinx Parameterized Macro, Version 2016.1
xpm_memory_dpdistram # (

    // Common module parameters
    .MEMORY_SIZE      (2048),           //positive integer
    .CLOCKING_MODE    ("common_clock"), //string; "common_clock", "independent_clock"
    .MEMORY_INIT_FILE ("none"),        //string; "none" or "<filename>.mem"
    .MEMORY_INIT_PARAM (""),           //string;
    .USE_MEM_INIT     (1),             //integer; 0,1
    .MESSAGE_CONTROL  (0),            //integer; 0,1

    // Port A module parameters
    .WRITE_DATA_WIDTH_A (32),          //positive integer
    .READ_DATA_WIDTH_A  (32),          //positive integer
    .BYTE_WRITE_WIDTH_A (32),          //integer; 8, 9, or WRITE_DATA_WIDTH_A value
    .ADDR_WIDTH_A       (6),           //positive integer
    .READ_RESET_VALUE_A ("0"),         //string
    .READ_LATENCY_A     (2),           //non-negative integer

    // Port B module parameters
    .READ_DATA_WIDTH_B  (32),          //positive integer
    .ADDR_WIDTH_B       (6),           //positive integer
    .READ_RESET_VALUE_B ("0"),         //string
    .READ_LATENCY_B     (2),           //non-negative integer

) xpm_memory_dpdistram_inst (

    // Port A module ports
    .clka  (clka),
    .rsta  (rsta),
    .ena   (ena),
    .regcea (regcea),
    .wea   (wea),
    .addra (addra),
    .dina  (dina),
    .douta (douta),

    // Port B module ports
    .clkb  (clkb),
    .rstb  (rstb),
    .enb   (enb),
    .regceb (regceb),
    .addrb (addrb),
    .doutb (doutb)

);

// End of xpm_memory_dpdistram instance declaration
```

## For More Information (individual items)

- See the [UltraScale Programmable Devices Documentation](#).
- Add more documents here if you want. Note that each architecture shown above will only show up in the appropriate Libraries Guide.

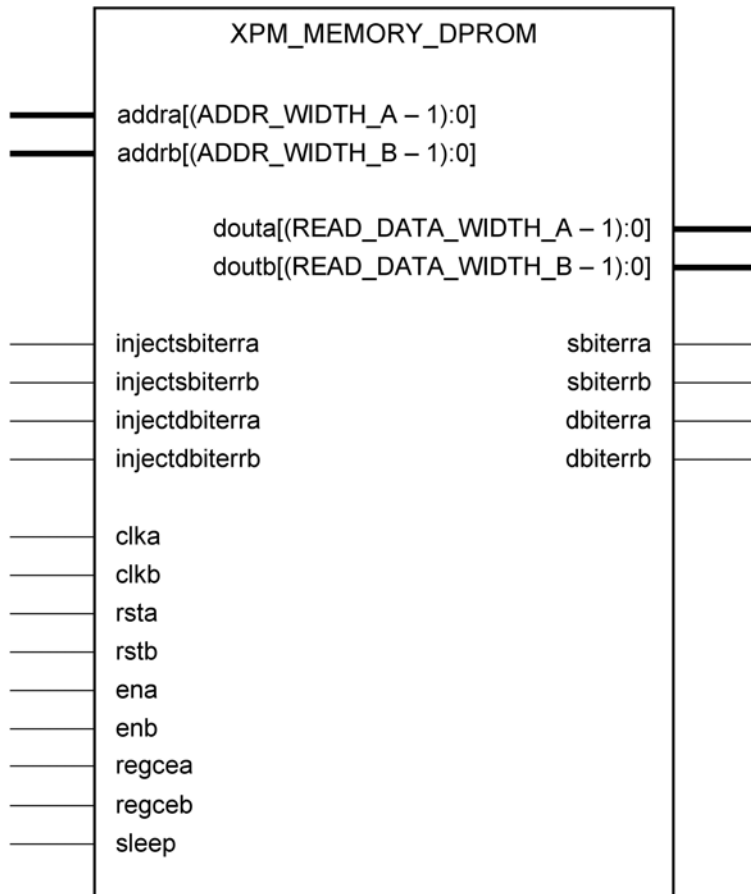
# XPM\_MEMORY\_DPRM

## Parameterized Macro: Dual Port ROM

MACRO\_GROUP: XPM

MACRO\_SUBGROUP: MEMORY

Families: UltraScale, UltraScale+



X16221-031116

## Introduction

This Macro is used to Instantiate True Port ROM. Read operation from the memory can be performed from both Port-A and Port-B simultaneously.

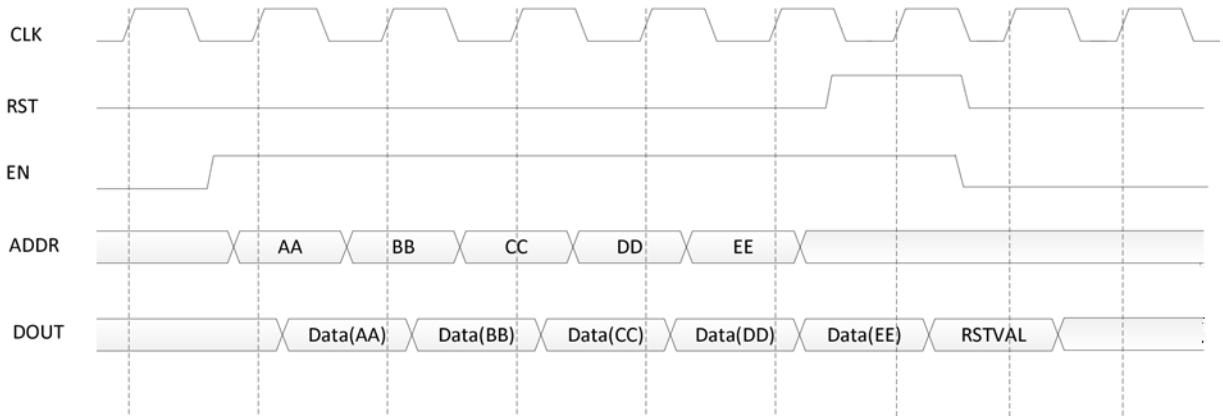
The following describes the basic read and write port usage of an XPM\_MEMORY instance. It does not distinguish between ports A and B.

- All synchronous signals are sensitive to the rising edge of clk[a|b], which is assumed to be a buffered and toggling clock signal behaving according to target device and memory primitive requirements.
- A read operation is implicitly performed to address addr[a|b] combinatorially. the data output is registered each clk[a|b] cycle that en[a|b] is asserted.
- Read data appears on the dout[a|b] port READ\_LATENCY\_[A|B] clk[a|b] cycles after the associated read operation.
- All read operations are gated by the value of en[a|b] on the initiating clk[a|b] cycle, regardless of input or output latencies.
- For each clk[a|b] cycle that rst[a|b] is asserted, the final output register is immediately but synchronously reset to READ\_RESET\_VALUE\_[A|B], irrespective of READ\_LATENCY\_[A|B].
- For each clk[a|b] cycle that regce[a|b] is asserted and rst[a|b] is de-asserted, the final output register captures and outputs the value from the previous pipeline register.
- Undriven or unknown values provided on module inputs will produce undefined memory array and output port behavior.

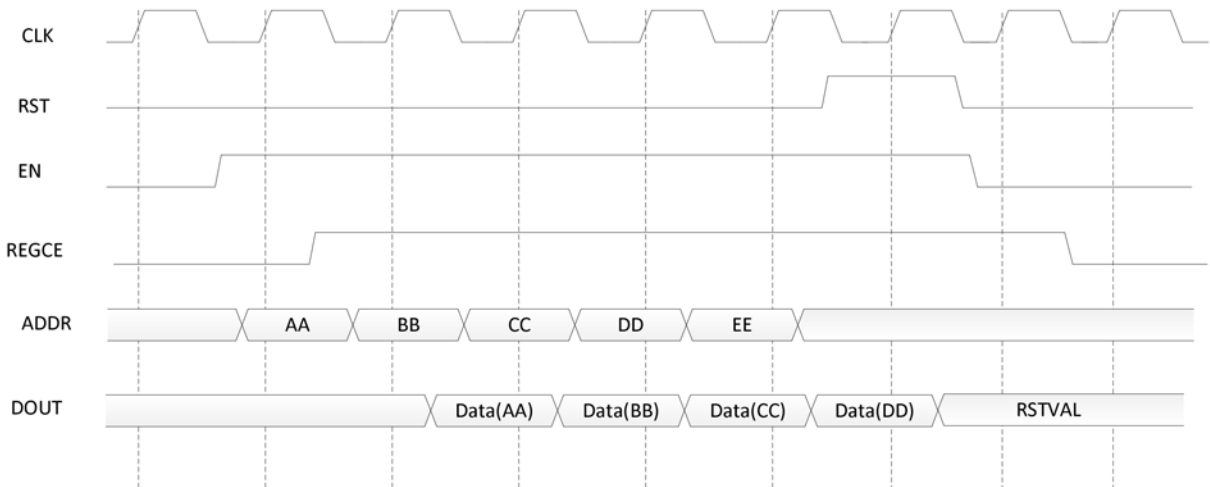
**NOTE:** When the attribute "CLOCKING\_MODE" is set to "common\_clock", then all the read/write operations to the memory through port-A and port-B are performed on clka. If this attribute is set to "independent\_clock", then read/write operations through port-A are performed based on clka, and read/write operations through port-B are performed based on clkb.

## Timing Diagrams

DPRROM with Read Latency of 1



DPRROM with Read Latency of 2



Note : The above waveforms does not distinguish between port-A and Port-B, the behavior listed in the above waveforms is true for both Port-A and Port-B

## Port Description

Port	Direction	Width	Sense	Handling if unused	Description
clka	Input	1	Rising-edge	Required	Port A Clock
clkb	Input	1	Rising-edge	Required	Port B Clock

Port	Direction	Width	Sense	Handling if unused	Description
rsta	Input	1	NA	Required	Port A final output register reset to READ_RESET_VALUE_A
rstb	Input	1	NA	Required	Port B final output register reset to READ_RESET_VALUE_B
ena	Input	1	NA	Required	Port A Enable
enb	Input	1	NA	Required	Port B Enable
regcea	Input	1	NA	Required	Port A final output pipeline register clock enable
regceb	Input	1	NA	Required	Port B final output pipeline register clock enable
addra	Input	ADDR_WIDTH_A	NA	Required	Port A Address
addrb	Input	ADDR_WIDTH_B	NA	Required	Port B Address
injectsbiterra	Input	1	NA	Tied to 0	Port A inject single bit error
injectsbiterrb	Input	1	NA	Tied to 0	Port B inject single bit error
injectdbiterra	Input	1	NA	Tied to 0	Port A inject double bit error
injectdbiterrb	Input	1	NA	Tied to 0	Port B inject double bit error
sleep	Input	1	NA	Tied to 0	Memory sleep control. <ul style="list-style-type: none"> <li>Synchronous to clka when CLOCKING_MODE is "common_clock"</li> <li>Asynchronous to clk[a b] when CLOCKING_MODE is "independent_clock"</li> </ul> <b>NOTE:</b> Sleep is being sampled on both clka and clkb ports in the behavioral simulations. Make sure that sleep is asserted in such a way that it gets detected by both clka and clkb.
douta	Output	READ_DATA_WIDTH_A	NA	Required	Port A output data
doutb	Output	READ_DATA_WIDTH_B	NA	Required	Port B output data
sbiterra	Output	1	NA	Tied to 0	Port A single bit error indicator
sbiterrb	Output	1	NA	Tied to 0	Port B single bit error indicator
dbiterra	Output	1	NA	Tied to 0	Port A double bit error indicator
dbiterrb	Output	1	NA	Tied to 0	Port B double bit error indicator

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
MEMORY_SIZE	Integer	>=1	2048	Total memory array size, in bits
MEMORY_PRIMITIVE	string	"auto", "distributed", "block", "ultra"	"auto"	Memory primitive type to map to: <ul style="list-style-type: none"> <li>"auto" = allows synthesis to choose memory primitive</li> <li>"distributed" = Distributed Memory</li> <li>"block" = Block Memory</li> <li>"ultra" = UltraRAM Memory</li> </ul>
CLOCKING_MODE	string	"common_clock", "independent_clock"	"common_clock"	Clocking Mode
READ_DATA_WIDTH_[A B]	Integer	>=1	32	Width of the port [A B] output data port, DOUT[A B]
READ_RESET_VALUE_[A B]	string	All type-legal values(hex string)	"0"	Initial and reset value of final register stage in port [A B] output data path
READ_LATENCY_[A B]	Integer	>=1	2	Number of register stages in the port [A B] read data path
MEMORY_INIT_FILE	Unspecified	"none", "<file_name>"	"none"	File name for memory content initialization.



Attribute	Type	Allowed Values	Default	Description
MEMORY_INIT_PARAM	Unspecified	"", valid hex String	""	<p>Initialize the contents of the memory using a parameter.</p> <p>Specify "" or "0" (including quotes) for no memory initialization through a parameter, or specify the string containing the hex characters. Each location must be separated by a delimiter (.).</p> <p>The parameter format must be ASCII and consist of only hexadecimal values organized into the specified depth by the narrowest data width generic value of the memory. For example, if the narrowest data width is 8, and the depth of memory is 8 locations, then the parameter value should be passed as shown below (0xAB in the below string is the 0th location and 0x78 is the 7th location).</p> <pre>MEMORY_INIT_PARAM = "AB,CD,EF,1,2,34,56,78"</pre> <p>Maximum supported memory size when MEMORY_INIT_PARAM is used is 4k bits.</p>
USE_MEMORY_INIT	Integer	0, 1	0	<ul style="list-style-type: none"> <li>1 = Enable the generation of below message</li> <li>0 = Disable the generation of below message</li> </ul> <p><b>NOTE:</b> This message gets generated only when there is no Memory Initialization specified either through file or parameter. INFO: MEMORY_INIT_FILE and MEMORY_INIT_PARAM together specifies no memory initialization. Initial memory contents will be all 0's</p>
WAKEUP_TIME	string	"disable_sleep", "use_sleep_pin"	"disable_sleep"	Number of clk[a] cycles to wake, following de-assertion of sleep. "use_sleep_pin" has wake up time of 2 clock cycles.
MESSAGE_CONTROL	Integer	0, 1	1	<p>Module messaging control</p> <ul style="list-style-type: none"> <li>0 = Disable dynamic messages (assertions)</li> <li>1 = Enable dynamic messages (assertions)</li> </ul>

## VHDL Instantiation Template

```
-- Copy the following two statements and paste them before the
-- entity declaration, unless they already exist.

Library xpm;
use xpm.vcomponents.all;

-- <--Cut the following instance declaration and paste it into
-- the architecture statement part of the design-->

-- xpm_memory_dprom: Dual Port ROM
-- Xilinx Parameterized Macro, Version 2016.1
xpm_memory_dprom_inst : xpm_memory_dprom
  generic map (
    -- Common module generics
    MEMORY_SIZE      => 2048,          --positive integer
    MEMORY_PRIMITIVE => "auto",        --string; "auto", "distributed", "block" or "ultra" ;
    CLOCKING_MODE    => "common_clock",--string; "common_clock", "independent_clock"
    MEMORY_INIT_FILE => "none",        --string; "none" or "<filename>.mem"
    MEMORY_INIT_PARAM => "",          --string;
    USE_MEM_INIT     => 1,            --integer; 0,1
    WAKEUP_TIME      => "disable_sleep",--string; "disable_sleep" or "use_sleep_pin"
    MESSAGE_CONTROL  => 0,            --integer; 0,1

    -- Port A module generics
    READ_DATA_WIDTH_A => 32,          --positive integer
    ADDR_WIDTH_A      => 6,           --positive integer
    READ_RESET_VALUE_A => "0",        --string
    READ_LATENCY_A    => 2,           --non-negative integer

    -- Port B module generics
    READ_DATA_WIDTH_B => 32,          --positive integer
    ADDR_WIDTH_B      => 6,           --positive integer
    READ_RESET_VALUE_B => "0",        --string
    READ_LATENCY_B    => 2,           --non-negative integer
  )
  port map (
    -- Common module ports
    sleep      => '0',

    -- Port A module ports
    clka       => clka,
    rsta       => rsta,
    ena        => ena,
    regcea     => regcea,
    addra      => addra,
    injectsbiterra => '0',    --do not change
    injectdbiterra => '0',    --do not change
    douta      => douta,
    sbiterra   => open,    --do not change
    dbiterra   => open,    --do not change

    -- Port B module ports
    clkb       => clkb,
    rstb       => rstb,
    enb        => enb,
    regceb     => regceb,
    addrb      => addrb,
    injectsbiterrb => '0',    --do not change
    injectdbiterrb => '0',    --do not change
    doutb      => doutb,
    sbiterrb   => open,    --do not change
    dbiterrb   => open,    --do not change
  );

-- End of xpm_memory_dprom_inst instance declaration
```

## Verilog Instantiation Template

```
// <--Cut the following instance declaration and paste it into
// the architecture statement part of the design-->

// xpm_memory_dprom: Dual Port ROM
// Xilinx Parameterized Macro, Version 2016.1
xpm_memory_dprom # (

    // Common module parameters
    .MEMORY_SIZE      (2048),           //positive integer
    .MEMORY_PRIMITIVE ("auto"),        //string; "auto", "distributed", "block" or "ultra";
    .CLOCKING_MODE    ("common_clock"), //string; "common_clock", "independent_clock"
    .MEMORY_INIT_FILE ("none"),        //string; "none" or "<filename>.mem"
    .MEMORY_INIT_PARAM (""),           //string;
    .USE_MEM_INIT     (1),             //integer; 0,1
    .WAKEUP_TIME      ("disable_sleep"), //string; "disable_sleep" or "use_sleep_pin"
    .MESSAGE_CONTROL  (0),             //integer; 0,1

    // Port A module parameters
    .READ_DATA_WIDTH_A (32),           //positive integer
    .ADDR_WIDTH_A      (6),           //positive integer
    .READ_RESET_VALUE_A ("0"),        //string
    .READ_LATENCY_A    (2),           //non-negative integer

    // Port B module parameters
    .READ_DATA_WIDTH_B (32),           //positive integer
    .ADDR_WIDTH_B      (6),           //positive integer
    .READ_RESET_VALUE_B ("0"),        //string
    .READ_LATENCY_B    (2),           //non-negative integer

) xpm_memory_dprom_inst (

    // Common module ports
    .sleep      (1'b0),

    // Port A module ports
    .clk_a      (clk_a),
    .rst_a      (rst_a),
    .ena        (ena),
    .regcea     (regcea),
    .addra      (addra),
    .injectsbiterra (1'b0), //do not change
    .injectdbiterra (1'b0), //do not change
    .douta      (douta),
    .sbiterra   (), //do not change
    .dbiterra   (), //do not change

    // Port B module ports
    .clk_b      (clk_b),
    .rst_b      (rst_b),
    .enb        (enb),
    .regceb     (regceb),
    .addrb      (addrb),
    .injectsbiterrb (1'b0), //do not change
    .injectdbiterrb (1'b0), //do not change
    .doutb      (doutb),
    .sbiterrb   (), //do not change
    .dbiterrb   (), //do not change

);

// End of xpm_memory_dprom instance declaration
```

## For More Information (individual items)

See the [UltraScale Programmable Devices Documentation](#).

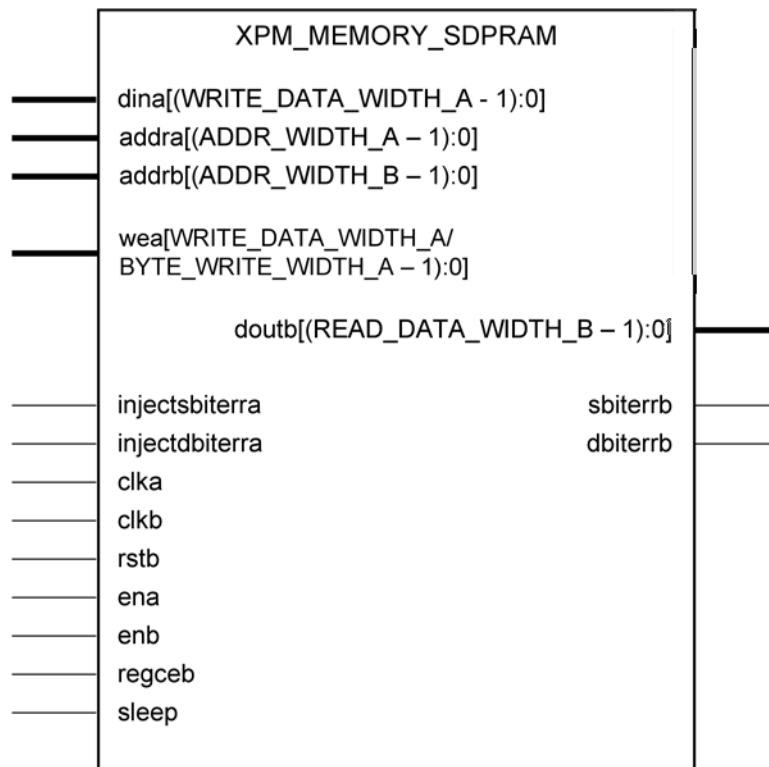
# XPM\_MEMORY\_SDPRAM

## Parameterized Macro: Simple Dual Port RAM

MACRO\_GROUP: XPM

MACRO\_SUBGROUP: MEMORY

Families: UltraScale, UltraScale+



X16233-033016

## Introduction

This Macro is used to Instantiate Simple Dual Port RAM. Port-A is used to perform write Operation to the memory, Port-B can be used to Read from the memory.

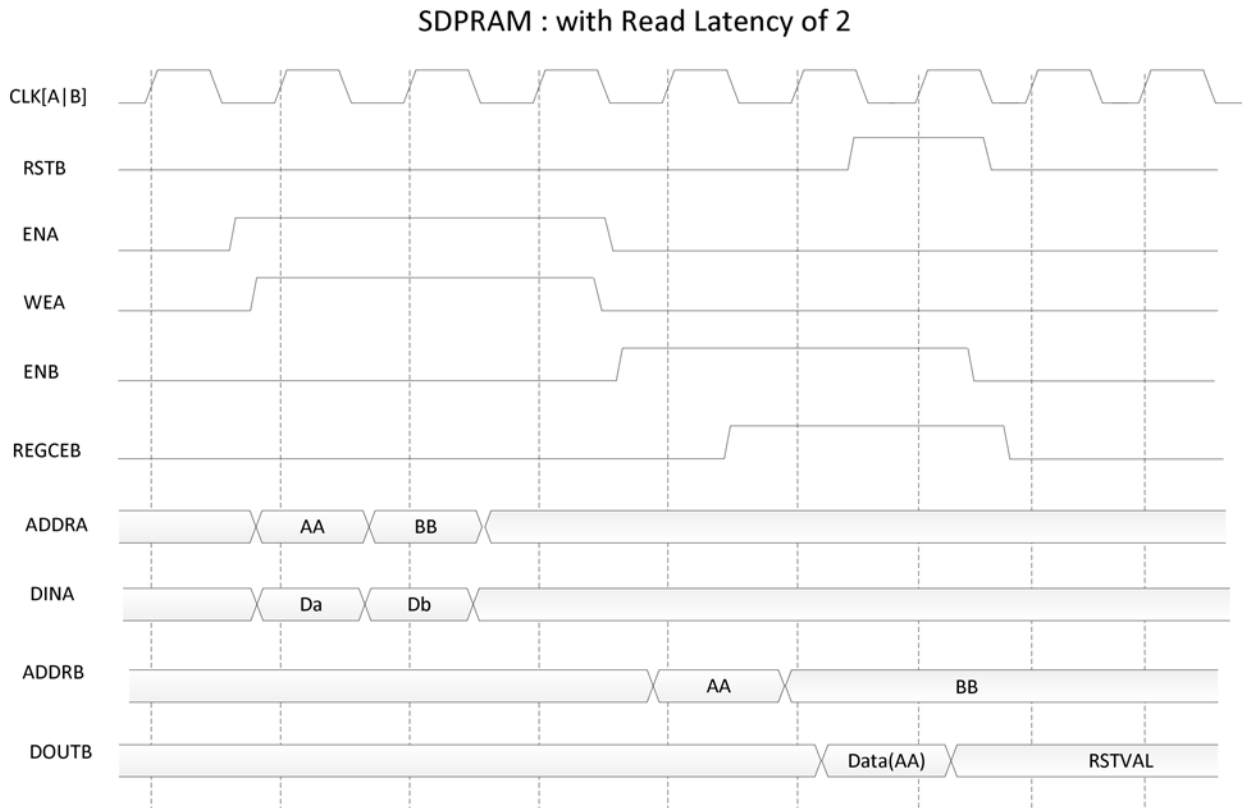
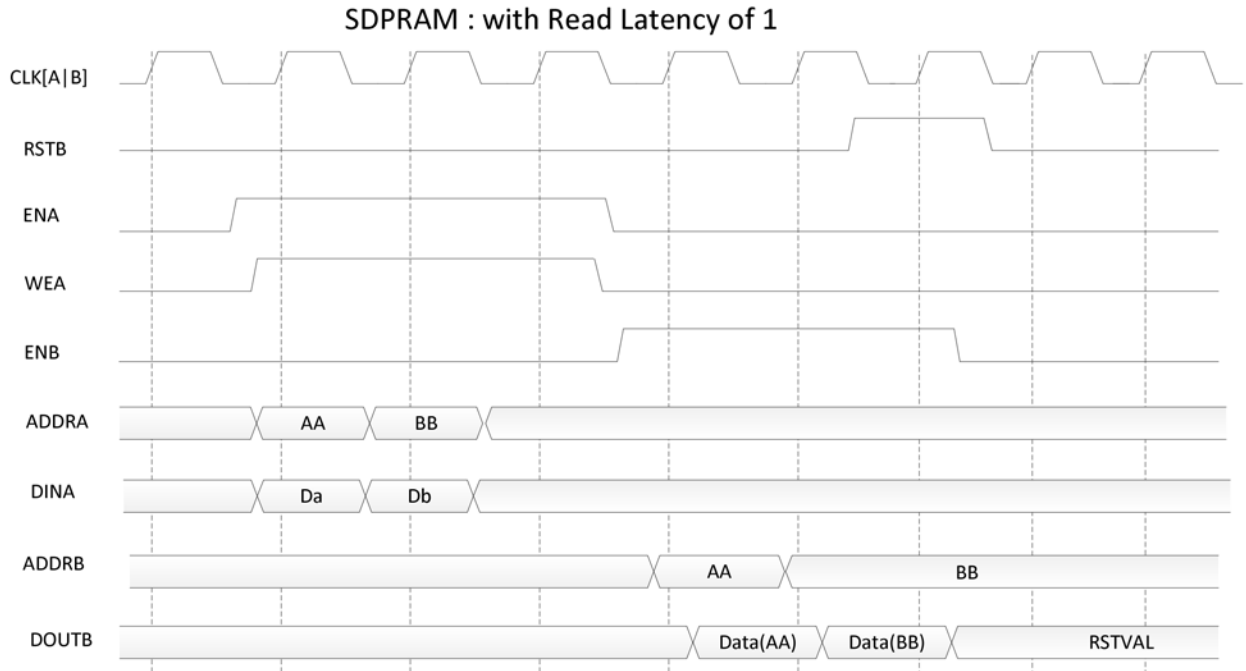
The following describes the basic read and write port usage of an XPM\_MEMORY instance. It does not distinguish between ports A and B.

- All synchronous signals are sensitive to the rising edge of clk[a|b], which is assumed to be a buffered and toggling clock signal behaving according to target device and memory primitive requirements.
- A read operation is implicitly performed to address addr[b] combinatorially. The data output is registered each clk[b] cycle that en[b] is asserted.
- Read data appears on the dout[b] port READ\_LATENCY\_[B] clk[b] cycles after the associated read operation.
- A write operation is explicitly performed, writing din[a] to address addr[a], when both en[a] and we[a] are asserted on each clk[a] cycle.

- All read and write operations are gated by the value of `en[a|b]` on the initiating `clk[a|b]` cycle, regardless of input or output latencies. The `addr[a]` and `we[a]` inputs have no effect when `en[a]` is de-asserted on the coincident `clk[a|b]` cycle.
- For each `clk[b]` cycle that `rst[b]` is asserted, the final output register is immediately but synchronously reset to `READ_RESET_VALUE_[B]`, irrespective of `READ_LATENCY_[B]`.
- For each `clk[b]` cycle that `regce[b]` is asserted and `rst[b]` is de-asserted, the final output register captures and outputs the value from the previous pipeline register.
- Undriven or unknown values provided on module inputs will produce undefined memory array and output port behavior.

**NOTE:** When the attribute "CLOCKING\_MODE" is set to "common\_clock", then all the read/write operations to the memory through port-A and port-B are performed on `clka`. If this attribute is set to "independent\_clock", then read/write operations through port-A are performed based on `clka`, and read/write operations through port-B are performed based on `clkb`.

## Timing Diagrams



## Port Description

Port	Direction	Width	Sense	Handling if unused	Description
clka	Input	1	Rising-edge	Required	Port A Clock
clkb	Input	1	Rising-edge	Required	Port B Clock
rstb	Input	1	NA	Required	Port B final output register reset to READ_RESET_VALUE_B
ena	Input	1	NA	Required	Port A Enable
enb	Input	1	NA	Required	Port B Enable
regceb	Input	1	NA	Required	Port B final output pipeline register clock enable
wea	Input	WRITE_DATA_WIDTH_A / BYTE_WRITE_WIDTH_A	NA	Required	Port A Write Enable
addra	Input	ADDR_WIDTH_A	NA	Required	Port A Address
addrb	Input	ADDR_WIDTH_B	NA	Required	Port B Address
dina	Input	WRITE_DATA_WIDTH_A	NA	Required	Port A Write Data
injectsbterra	Input	1	NA	Tied to 0	Port A inject single bit error
injectdbterra	Input	1	NA	Tied to 0	Port A inject double bit error
sleep	Input	1	NA	Tied to 0	Memory sleep control. <ul style="list-style-type: none"> <li>Synchronous to clka when CLOCKING_MODE is "common_clock"</li> <li>Asynchronous to clk[a]b when CLOCKING_MODE is "independent_clock"</li> </ul> <b>NOTE:</b> Sleep is being sampled on both clka and clkb ports in the behavioral simulations. Make sure that sleep is asserted in such a way that it gets detected by both clka and clkb.
doutb	Output	READ_DATA_WIDTH_B	NA	Required	Port B output data
sbiterrb	Output	1	NA	Tied to 0	Port B single bit error indicator
dbiterrb	Output	1	NA	Tied to 0	Port B double bit error indicator

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
MEMORY_SIZE	Integer	>=1	2048	Total memory array size, in bits.
MEMORY_PRIMITIVE	string	"auto", "distributed", "block", "ultra"	"auto"	Memory primitive type to map to. <ul style="list-style-type: none"> <li>"auto" = allows synthesis to choose memory primitive</li> <li>"distributed" = Distributed Memory</li> <li>"block" = Block Memory</li> <li>"ultra" = UltraRAM Memory</li> </ul>
CLOCKING_MODE	string	"common_clock", "independent_clock"	"common_clock"	Clocking Mode.
WRITE_DATA_WIDTH_[A]	Integer	>=1	32	Width of the port [A] write data port, din[a].
BYTE_WRITE_WIDTH_[A]	Integer	8, 9, WRITE_DATA_WIDTH_[A]	WRITE_DATA_WIDTH_[A]	Width of each byte, in bits, for enabled byte-wide write functionality of port [A]
READ_DATA_WIDTH_[A B]	Integer	>=1	WRITE_DATA_WIDTH_[A]	Width of the port [A] output data port, dout[a].
READ_RESET_VALUE_[B]	string	All type-legal values (hex string)	"0"	Initial and reset value of final register stage in port [B] output data path.
READ_LATENCY_[B]	Integer	>=1	2	Number of register stages in the port [B] read data path.
WRITE_MODE_[B]	string	"write_first", "read_first", "no_change"	"no_change"	Write mode behavior for port [A B].
USE_MEMORY_INIT	Integer	0, 1	0	<ul style="list-style-type: none"> <li>1 = Enable the generation of below message</li> <li>0 = Disable the generation of below message</li> </ul> <p><b>NOTE:</b> This message gets generated only when there is no Memory Initialization specified either through file or parameter. INFO: MEMORY_INIT_FILE and MEMORY_INIT_PARAM together specifies no memory initialization. Initial memory contents will be all 0's</p>
MEMORY_INIT_FILE	Unspecified	"none", "<file_name>"	"none"	File name for memory content initialization.



Attribute	Type	Allowed Values	Default	Description
MEMORY_INIT_PARAM	Unspecified	"" , valid hex String	""	<p>Initialize the contents of the memory using a parameter.</p> <p>Specify "" or "0" (including quotes) for no memory initialization through a parameter, or specify the string containing the hex characters. Each location must be separated by a delimiter (,).</p> <p>The parameter format must be ASCII and consist of only hexadecimal values organized into the specified depth by the narrowest data width generic value of the memory. For example, if the narrowest data width is 8, and the depth of memory is 8 locations, then the parameter value should be passed as shown below (0xAB in the below string is the 0th location and 0x78 is the 7th location).</p> <p>MEMORY_INIT_PARAM = "AB,CD,EF,1,2,34,56,78"</p> <p>Maximum supported memory size when MEMORY_INIT_PARAM is used is 4k bits.</p>
WAKEUP_TIME	String	"disable_sleep", "use_sleep_pin"	"disable_sleep"	<p>Number of clk[a] cycles to wake, following de-assertion of sleep. "use_sleep_pin" has wake up time of 2 clock cycles.</p>
MESSAGE_CONTROL	Integer	0, 1	1	<p>Module messaging control.</p> <ul style="list-style-type: none"> <li>0 = Disable dynamic messages (assertions)</li> <li>1 = Enable dynamic messages (assertions)</li> </ul>

## VHDL Instantiation Template

```
-- Copy the following two statements and paste them before the
-- entity declaration, unless they already exist.
```

```
Library xpm;
use xpm.vcomponents.all;
```

```
-- <--Cut the following instance declaration and paste it into
-- the architecture statement part of the design-->
```

```
-- xpm_memory_sdpram: Simple Dual Port RAM
-- Xilinx Parameterized Macro, Version 2016.1
xpm_memory_sdpram_inst : xpm_memory_sdpram
  generic map (
```

```
    -- Common module generics
    MEMORY_SIZE      => 2048,          --positive integer
    MEMORY_PRIMITIVE => "auto",       --string; "auto", "distributed", "block" or "ultra" ;
    CLOCKING_MODE    => "common_clock",--string; "common_clock", "independent_clock"
```

```

MEMORY_INIT_FILE    => "none",          --string; "none" or "<filename>.mem"
MEMORY_INIT_PARAM  => "",              --string;
USE_MEM_INIT        => 1,              --integer; 0,1
WAKEUP_TIME        => "disable_sleep",--string; "disable_sleep" or "use_sleep_pin"
MESSAGE_CONTROL    => 0,              --integer; 0,1

-- Port A module generics
WRITE_DATA_WIDTH_A => 32,              --positive integer
BYTE_WRITE_WIDTH_A => 32,              --integer; 8, 9, or WRITE_DATA_WIDTH_A value
ADDR_WIDTH_A       => 6,              --positive integer

-- Port B module generics
READ_DATA_WIDTH_B  => 32,              --positive integer
ADDR_WIDTH_B       => 6,              --positive integer
READ_RESET_VALUE_B => "0",            --string
READ_LATENCY_B     => 2,              --non-negative integer
WRITE_MODE_B       => "no_change"     --string; "write_first", "read_first", "no_change"
)
port map (

-- Common module ports
sleep          => '0',

-- Port A module ports
clk_a          => clk_a,
ena            => ena,
wea           => wea,
addra         => addra,
dina          => dina,
injectsbiterra => '0', --do not change
injectdbiterra => '0', --do not change

-- Port B module ports
clk_b          => clk_b,
rstb           => rstb,
enb            => enb,
regceb         => regceb,
addrb          => addrb,
doutb          => doutb,
sbiterrb       => open, --do not change
dbiterrb       => open, --do not change
);

-- End of xpm_memory_sdpram_inst instance declaration

```

## Verilog Instantiation Template

```
// <--Cut the following instance declaration and paste it into
// the architecture statement part of the design-->

// xpm_memory_sdpram: Simple Dual Port RAM
// Xilinx Parameterized Macro, Version 2016.1
xpm_memory_sdpram # (

    // Common module parameters
    .MEMORY_SIZE      (2048),           //positive integer
    .MEMORY_PRIMITIVE ("auto"),        //string; "auto", "distributed", "block" or "ultra";
    .CLOCKING_MODE    ("common_clock"), //string; "common_clock", "independent_clock"
    .MEMORY_INIT_FILE ("none"),        //string; "none" or "<filename>.mem"
    .MEMORY_INIT_PARAM (""),          //string;
    .USE_MEM_INIT     (1),             //integer; 0,1
    .WAKEUP_TIME      ("disable_sleep"), //string; "disable_sleep" or "use_sleep_pin"
    .MESSAGE_CONTROL  (0),            //integer; 0,1

    // Port A module parameters
    .WRITE_DATA_WIDTH_A (32),          //positive integer
    .BYTE_WRITE_WIDTH_A (32),         //integer; 8, 9, or WRITE_DATA_WIDTH_A value
    .ADDR_WIDTH_A      (6),           //positive integer

    // Port B module parameters
    .READ_DATA_WIDTH_B (32),          //positive integer
    .ADDR_WIDTH_B      (6),           //positive integer
    .READ_RESET_VALUE_B ("0"),        //string
    .READ_LATENCY_B    (2),           //non-negative integer
    .WRITE_MODE_B      ("no_change")  //string; "write_first", "read_first", "no_change"

) xpm_memory_sdpram_inst (

    // Common module ports
    .sleep      (1'b0),

    // Port A module ports
    .clka      (clka),
    .ena       (ena),
    .wea       (wea),
    .addra     (addra),
    .dina      (dina),
    .injectsbiterra (1'b0), //do not change
    .injectdbiterra (1'b0), //do not change

    // Port B module ports
    .clkb      (clkb),
    .rstb      (rstb),
    .enb       (enb),
    .regceb    (regceb),
    .addrb     (addrb),
    .doutb     (doutb),
    .sbiterrb  (), //do not change
    .dbiterrb  (), //do not change

);

// End of xpm_memory_sdpram instance declaration
```

## For More Information (individual items)

See the [UltraScale Programmable Devices Documentation](#).

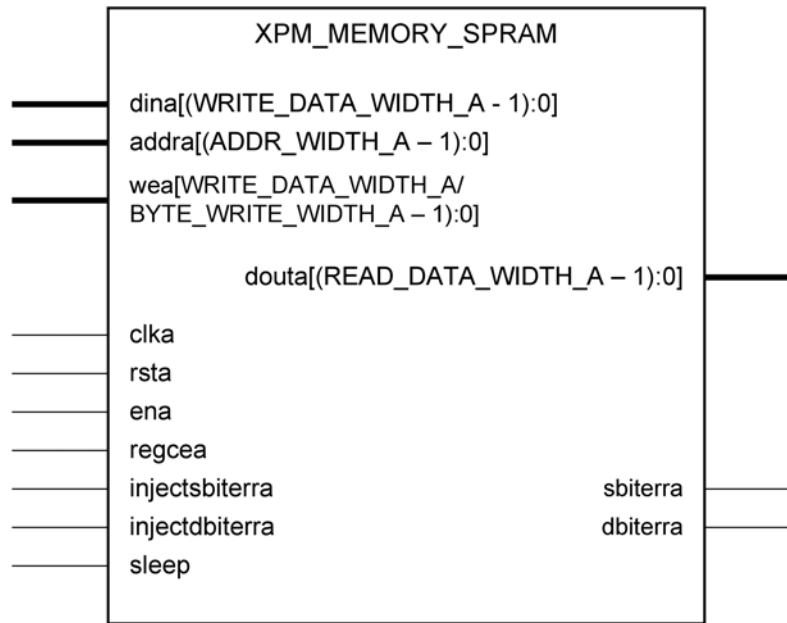
# XPM\_MEMORY\_SPRAM

## Parameterized Macro: Single Port RAM

MACRO\_GROUP: XPM

MACRO\_SUBGROUP: MEMORY

Families: UltraScale, UltraScale+



X16218-031116

## Introduction

This Macro is used to Instantiate single Port RAM. Read and writes to the memory can be done through Port-A.

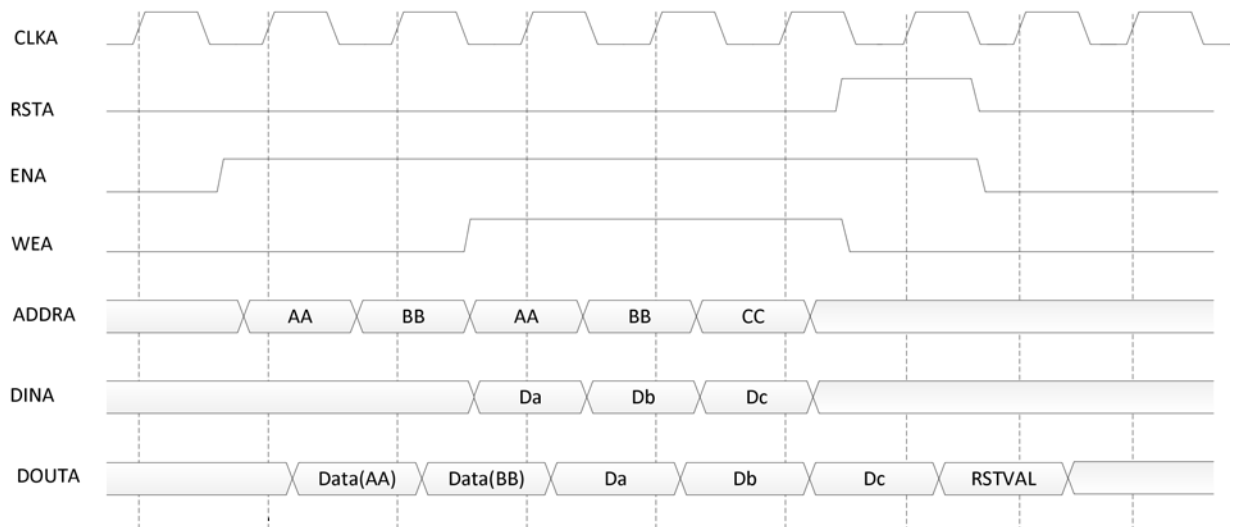
The following describes the basic read and write port usage of an XPM\_MEMORY instance.

- All synchronous signals are sensitive to the rising edge of `clka`, which is assumed to be a buffered and toggling clock signal behaving according to target device and memory primitive requirements.
- A read operation is implicitly performed to address `ADDRA` combinatorially. The data output is registered each `CLKA` cycle that `ENA` is asserted.
- Read data appears on the `douta` port `READ_LATENCY_A` `clka` cycles after the associated read operation.
- A write operation is explicitly performed, writing `dina` to address `addra`, when both `ena` and `wea` are asserted on each `clka` cycle.
- All read and write operations are gated by the value of `ena` on the initiating `clka` cycle, regardless of input or output latencies. the `addra` and `wea` inputs have no effect when `ena` is de-asserted on the coincident `clka` cycle.
- The behavior of `douta` with respect to the combination of `dina` and `addra` is a function of `WRITE_MODE_A`.

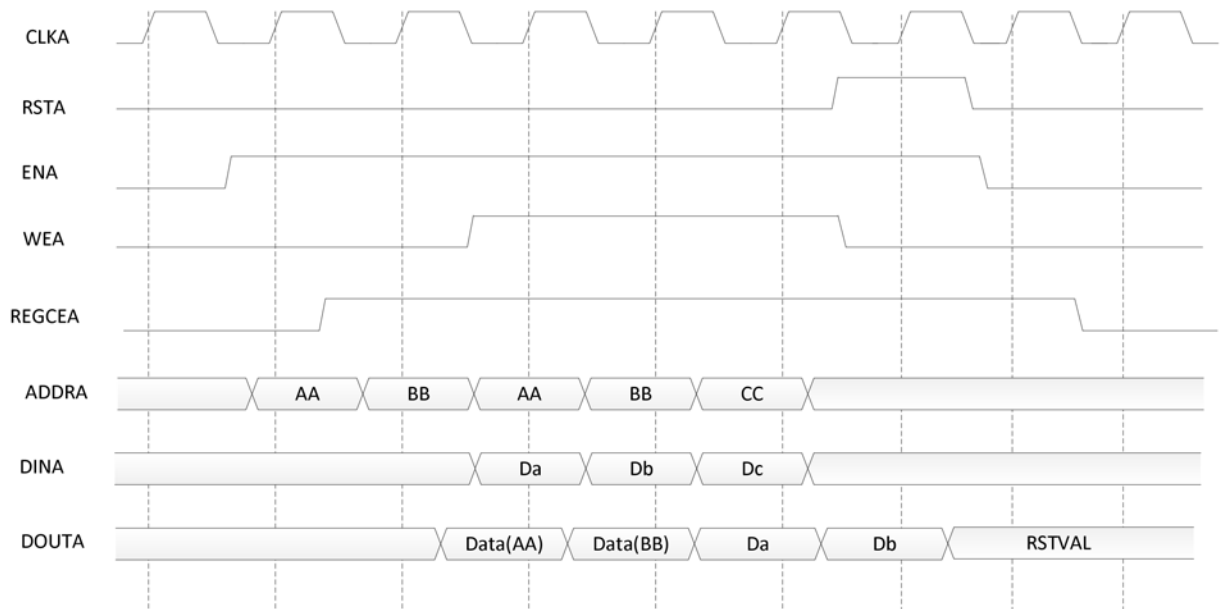
- For each *clka* cycle that *rsta* is asserted, the final output register is immediately but synchronously reset to `READ_RESET_VALUE_A`, irrespective of `READ_LATENCY_A`.
- For each *clka* cycle that *regcea* is asserted and *rsta* is de-asserted, the final output register captures and outputs the value from the previous pipeline register.
- Undriven or unknown values provided on module inputs will produce undefined memory array and output port behavior.

## Timing Diagrams

SPRAM : Write First Mode with Read Latency of 1



SPRAM : Write First Mode with Read Latency of 2



## Port Description

Port	Direction	Width	Sense	Handling if unused	Description
clka	Input	1	Rising-edge	Required	Port A Clock
rsta	Input	1	NA	Required	Port A final output register reset to READ_RESET_VALUE_A
ena	Input	1	NA	Required	Port A Enable
regcea	Input	1	NA	Required	Port A final output pipeline register clock enable
wea	Input	WRITE_DATA_WIDTH_A / BYTE_WRITE_WIDTH_A	NA	Required	Port A Write Enable
addra	Input	ADDR_WIDTH_A	NA	Required	Port A Address
dina	Input	WRITE_DATA_WIDTH_A	NA	Required	Port A Write Data
injectsbterra	Input	1	NA	Tied to 0	Port A inject single bit error
injectdbterra	Input	1	NA	Tied to 0	Port A inject double bit error
sleep	Input	1	NA	Tied to 0	Memory sleep control. <ul style="list-style-type: none"> <li>Synchronous to clka when CLOCKING_MODE is "common_clock"</li> <li>Asynchronous to clk[a b] when CLOCKING_MODE is "independent_clock"</li> </ul> <b>NOTE:</b> Sleep is being sampled on both clka and clkb ports in the behavioral simulations. Make sure that sleep is asserted in such a way that it gets detected by both clka and clkb.
douta	Output	READ_DATA_WIDTH_A	NA	Required	Port A output data
sbiterra	Output	1	NA	Tied to 0	Port A single bit error indicator
dbterra	Output	1	NA	Tied to 0	Port A double bit error indicator

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
MEMORY_SIZE	Integer	>=1	2048	Total memory array size, in bits
MEMORY_PRIMITIVE	string	"auto", "distributed", "block", "ultra"	"auto"	Memory primitive type to map to: <ul style="list-style-type: none"> <li>"auto" = allows synthesis to choose memory primitive</li> <li>"distributed" = Distributed Memory</li> <li>"block" = Block Memory</li> <li>"ultra" = UltraRAM Memory</li> </ul>
WRITE_DATA_WIDTH_[A]	Integer	>=1	32	Width of the port [A] write data port, din[a]
BYTE_WRITE_WIDTH_[A]	Integer	8, 9, WRITE_DATA_WIDTH_[A]	WRITE_DATA_WIDTH_[A]	Width of each byte, in bits, for enabled byte-wide write functionality of port [A]
READ_DATA_WIDTH_[A]	Integer	>=1	WRITE_DATA_WIDTH_[A]	Width of the port [A] output data port, dout[a]
READ_RESET_VALUE_[A]	string	All type-legal values (hex string)	"0"	Initial and reset value of final register stage in port [A] output data path
READ_LATENCY_[A]	Integer	>=1	2	Number of register stages in the port [A] read data path
WRITE_MODE_[A]	string	"write_first", "read_first", "no_change"	"no_change"	Write mode behavior for port [A] output data port, dout[a]
USE_MEMORY_INIT	Integer	0, 1	0	<ul style="list-style-type: none"> <li>1 = Enable the generation of below message</li> <li>0 = Disable the generation of below message</li> </ul> <p><b>NOTE:</b> This message gets generated only when there is no Memory Initialization specified either through file or parameter. INFO: MEMORY_INIT_FILE and MEMORY_INIT_PARAM together specifies no memory initialization. Initial memory contents will be all 0's</p>
MEMORY_INIT_FILE	Unspecified	"none", "<file_name>"	"none"	File name for memory content initialization.

Attribute	Type	Allowed Values	Default	Description
MEMORY_INIT_PARAM	Unspecified	"", valid hex String	""	<p>Initialize the contents of the memory using a parameter.</p> <p>Specify "" or "0" (including quotes) for no memory initialization through a parameter, or specify the string containing the hex characters. Each location must be separated by a delimiter (,).</p> <p>The parameter format must be ASCII and consist of only hexadecimal values organized into the specified depth by the narrowest data width generic value of the memory. For example, if the narrowest data width is 8, and the depth of memory is 8 locations, then the parameter value should be passed as shown below (0xAB in the below string is the 0th location and 0x78 is the 7th location).</p> <p>MEMORY_INIT_PARAM = "AB,CD,EF,1,2,34,56,78"</p> <p>Maximum supported memory size when MEMORY_INIT_PARAM is used is 4k bits.</p>
WAKEUP_TIME	string	"disable_sleep", "use_sleep_pin"	"disable_sleep"	<p>Number of clk[a] cycles to wake, following de-assertion of sleep. "use_sleep_pin" has wake up time of 2 clock cycles.</p>
MESSAGE_CONTROL	Integer	0, 1	1	<p>Module messaging control</p> <ul style="list-style-type: none"> <li>0 = Disable dynamic messages (assertions)</li> <li>1 = Enable dynamic messages (assertions)</li> </ul>

## VHDL Instantiation Template

```

-- Copy the following two statements and paste them before the
-- entity declaration, unless they already exist.

Library xpm;
use xpm.vcomponents.all;

-- <--Cut the following instance declaration and paste it into
-- the architecture statement part of the design-->

-- xpm_memory_spram: Single Port RAM
-- Xilinx Parameterized Macro, Version 2016.1
xpm_memory_spram_inst : xpm_memory_spram
  generic map (
    -- Common module generics
    MEMORY_SIZE      => 2048,          --positive integer
    MEMORY_PRIMITIVE => "auto",       --string; "auto", "distributed", "block" or "ultra" ;
    MEMORY_INIT_FILE => "none",      --string; "none" or "<filename>.mem"
    MEMORY_INIT_PARAM => "",         --string;
    USE_MEM_INIT     => 1,           --integer; 0,1
  )

```



```

WAKEUP_TIME      => "disable_sleep",--string; "disable_sleep" or "use_sleep_pin"
MESSAGE_CONTROL  => 0,                --integer; 0,1

-- Port A module generics
WRITE_DATA_WIDTH_A => 32,            --positive integer
READ_DATA_WIDTH_A  => 32,            --positive integer
BYTE_WRITE_WIDTH_A => 32,            --integer; 8, 9, or WRITE_DATA_WIDTH_A value
ADDR_WIDTH_A       => 6,             --positive integer
READ_RESET_VALUE_A => "0",           --string
READ_LATENCY_A     => 2,             --non-negative integer
WRITE_MODE_A       => "read_first" --string; "write_first", "read_first", "no_change"
)
port map (

-- Common module ports
sleep      => '0',

-- Port A module ports
clka       => clka,
rsta       => rsta,
ena        => ena,
regcea     => regcea,
wea        => wea,
addra      => addra,
dina       => dina,
injectsbiterra => '0', --do not change
injectdbiterra => '0', --do not change
douta      => douta,
sbiterra   => open, --do not change
dbiterra   => open --do not change
);

-- End of xpm_memory_spram_inst instance declaration

```

## Verilog Instantiation Template

```
// <--Cut the following instance declaration and paste it into
// the architecture statement part of the design-->

// xpm_memory_spram: Single Port RAM
// Xilinx Parameterized Macro, Version 2016.1
xpm_memory_spram # (

    // Common module parameters
    .MEMORY_SIZE      (2048),           //positive integer
    .MEMORY_PRIMITIVE ("auto"),         //string; "auto", "distributed", "block" or "ultra";
    .MEMORY_INIT_FILE ("none"),         //string; "none" or "<filename>.mem"
    .MEMORY_INIT_PARAM (""),           //string;
    .USE_MEM_INIT     (1),              //integer; 0,1
    .WAKEUP_TIME       ("disable_sleep"), //string; "disable_sleep" or "use_sleep_pin"
    .MESSAGE_CONTROL   (0),             //integer; 0,1

    // Port A module parameters
    .WRITE_DATA_WIDTH_A (32),           //positive integer
    .READ_DATA_WIDTH_A  (32),           //positive integer
    .BYTE_WRITE_WIDTH_A (32),           //integer; 8, 9, or WRITE_DATA_WIDTH_A value
    .ADDR_WIDTH_A       (6),            //positive integer
    .READ_RESET_VALUE_A ("0"),          //string
    .READ_LATENCY_A     (2),            //non-negative integer
    .WRITE_MODE_A       ("read_first")  //string; "write_first", "read_first", "no_change"

) xpm_memory_spram_inst (

    // Common module ports
    .sleep      (1'b0),

    // Port A module ports
    .clka       (clka),
    .rsta       (rsta),
    .ena        (ena),
    .regcea     (regcea),
    .wea        (wea),
    .addra      (addra),
    .dina       (dina),
    .injectsbiterra (1'b0), //do not change
    .injectdbiterra (1'b0), //do not change
    .douta      (douta),
    .sbiterra   (),         //do not change
    .dbiterra   ()         //do not change

);

// End of xpm_memory_spram instance declaration
```

## For More Information (individual items)

- See the [UltraScale Programmable Devices Documentation](#).
- Add more documents here if you want. Note that each architecture shown above will only show up in the appropriate Libraries Guide.

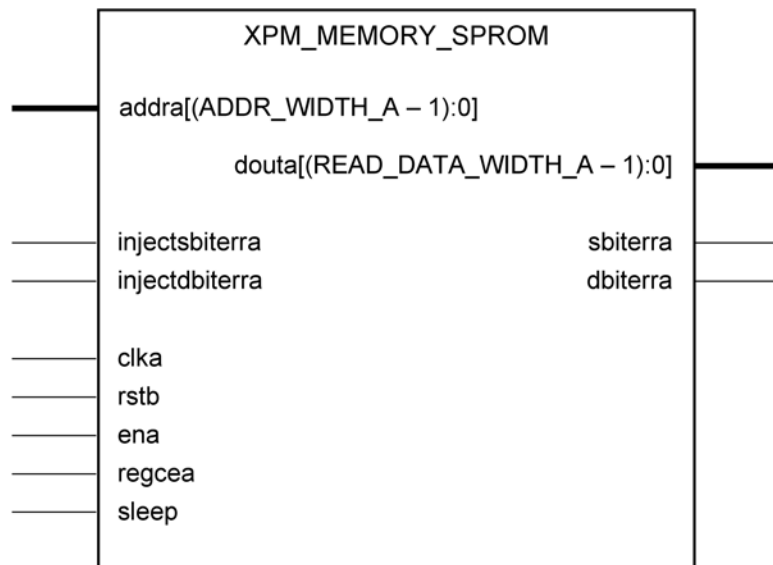
# XPM\_MEMORY\_SPROM

## Parameterized Macro: Single Port ROM

MACRO\_GROUP: XPM

MACRO\_SUBGROUP: MEMORY

Families: UltraScale, UltraScale+



X16250-031116

## Introduction

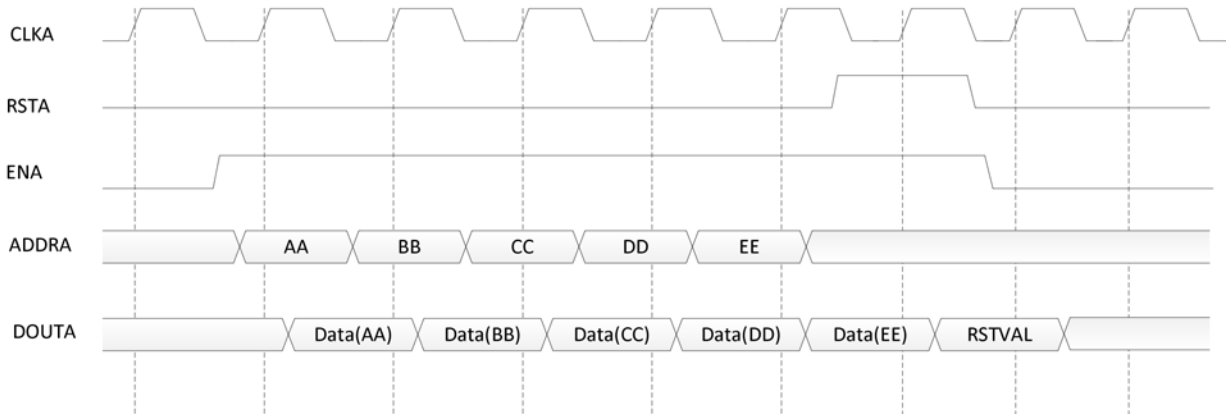
This Macro is used to Instantiate single Port ROM. Read operation from the memory can be performed through Port-A. No Write Operations are allowed.

The following describes the basic read and write port usage of an XPM\_MEMORY instance.

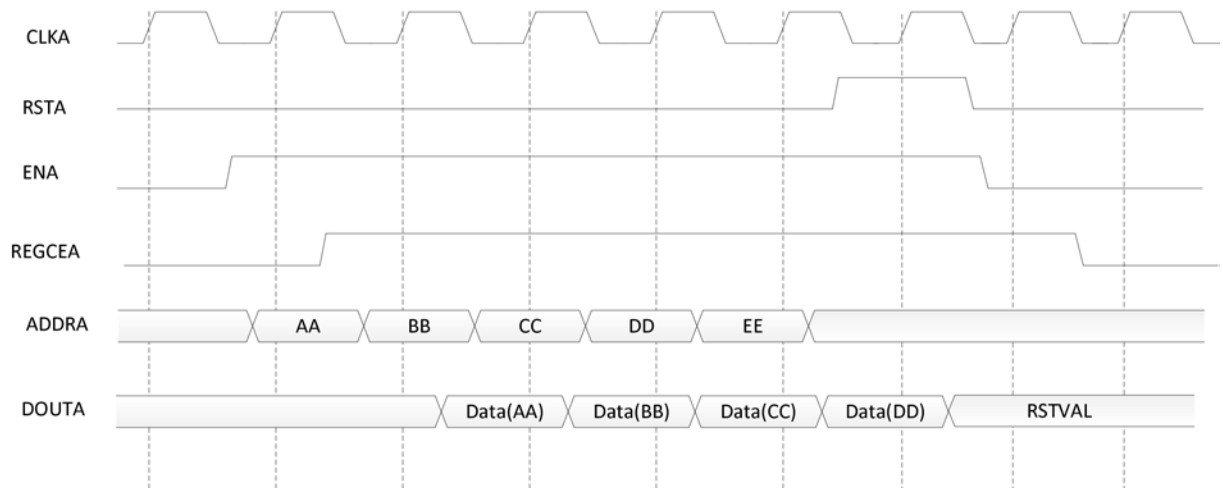
- All synchronous signals are sensitive to the rising edge of clka, which is assumed to be a buffered and toggling clock signal behaving according to target device and memory primitive requirements.
- A read operation is implicitly performed to address ADDRA combinatorially. The data output is registered each CLKA cycle that ena is asserted.
- Read data appears on the douta port READ\_LATENCY\_A CLKA cycles after the associated read operation.
- All read operations are gated by the value of ena on the initiating clka cycle, regardless of input or output latencies.
- For each clka cycle that rsta is asserted, the final output register is immediately but synchronously reset to READ\_RESET\_VALUE\_A, irrespective of READ\_LATENCY\_A.
- For each clka cycle that regcea is asserted and rsta is de-asserted, the final output register captures and outputs the value from the previous pipeline register.
- Undriven or unknown values provided on module inputs will produce undefined memory array and output port behavior.

## Timing Diagrams

SPROM with Read Latency of 1



SPROM with Read Latency of 2



## Port Description

Port	Direction	Width	Sense	Handling if unused	Description
clka	Input	1	Rising-edge	Required	Port A Clock
rsta	Input	1	NA	Required	Port A final output register reset to READ_RESET_VALUE_A
ena	Input	1	NA	Required	Port A Enable
regcea	Input	1	NA	Required	Port A final output pipeline register clock enable
addr_a	Input	ADDR_WIDTH_A	NA	Required	Port A Address

Port	Direction	Width	Sense	Handling if unused	Description
injectsbiterra	Input	1	NA	Tied to 0	Port A inject single bit error
injectdbiterra	Input	1	NA	Tied to 0	Port A inject double bit error
sleep	Input	1	NA	Required	Memory sleep control. <ul style="list-style-type: none"> <li>Synchronous to clka when CLOCKING_MODE is "common_clock"</li> <li>Asynchronous to clk[a b] when CLOCKING_MODE is "independent_clock"</li> </ul> <b>NOTE:</b> Sleep is being sampled on both clka and clkb ports in the behavioral simulations. Make sure that sleep is asserted in such a way that it gets detected by both clka and clkb.
douta	Output	READ_DATA_WIDTH_A	NA	Required	Port A output data
sbiterra	Output	1	NA	Tied to 0	Port A single bit error indicator
dbiterra	Output	1	NA	Tied to 0	Port A double bit error indicator

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
MEMORY_SIZE	Integer	>=1	2048	Total memory array size, in bits
MEMORY_PRIMITIVE	string	"auto", "distributed", "block", "ultra"	"auto"	Memory primitive type to map to: <ul style="list-style-type: none"> <li>"auto" = allows synthesis to choose memory primitive</li> <li>"distributed" = Distributed Memory</li> <li>"block" = Block Memory</li> <li>"ultra" = UltraRAM Memory</li> </ul>
READ_DATA_WIDTH_[A]	Integer	>=1	32	Width of the port [A] output data port, dout[a]
READ_RESET_VALUE_[A]	string	All type-legal values(hex string)	"0"	Initial and reset value of final register stage in port [A] output data path

Attribute	Type	Allowed Values	Default	Description
READ_LATENCY_[A]	Integer	>=1	2	Number of register stages in the port [A] read data path
USE_MEMORY_INIT	Integer	0, 1	0	<ul style="list-style-type: none"> <li>1 = Enable the generation of below message</li> <li>0 = Disable the generation of below message</li> </ul> <p><b>NOTE:</b> This message gets generated only when there is no Memory Initialization specified either through file or parameter.  <b>INFO:</b> MEMORY_INIT_FILE and MEMORY_INIT_PARAM together specifies no memory initialization. Initial memory contents will be all 0's</p>
MEMORY_INIT_FILE	Unspecified	"none", "<file_name>"	"none"	File name for memory content initialization.
MEMORY_INIT_PARAM	Unspecified	"", valid hex String	""	<p>Initialize the contents of the memory using a parameter.</p> <p>Specify "" or "0" (including quotes) for no memory initialization through a parameter, or specify the string containing the hex characters. Each location must be separated by a delimiter (.).</p> <p>The parameter format must be ASCII and consist of only hexadecimal values organized into the specified depth by the narrowest data width generic value of the memory. For example, if the narrowest data width is 8, and the depth of memory is 8 locations, then the parameter value should be passed as shown below (0xAB in the below string is the 0th location and 0x78 is the 7th location).</p> <p>MEMORY_INIT_PARAM = "AB,CD,EF,1,2,34,56,78"</p> <p>Maximum supported memory size when MEMORY_INIT_PARAM is used is 4k bits.</p>
WAKEUP_TIME	string	"disable_sleep", "use_sleep_pin"	"disable_sleep"	Number of clk[a] cycles to wake, following de-assertion of sleep. "use_sleep_pin" has wake up time of 2 clock cycles.

Attribute	Type	Allowed Values	Default	Description
VERSION	Integer	All type-legal values	0	Module version number
MESSAGE_CONTROL	Integer	0, 1	1	Module messaging control <ul style="list-style-type: none"> <li>0 = Disable dynamic messages (assertions)</li> <li>1 = Enable dynamic messages (assertions)</li> </ul>

## VHDL Instantiation Template

```

-- Copy the following two statements and paste them before the
-- entity declaration, unless they already exist.

Library xpm;
use xpm.vcomponents.all;

-- <--Cut the following instance declaration and paste it into
-- the architecture statement part of the design-->

-- xpm_memory_sprom: Single Port ROM
-- Xilinx Parameterized Macro, Version 2016.1
xpm_memory_sprom_inst : xpm_memory_sprom
    generic map (
        -- Common module generics
        MEMORY_SIZE      => 2048,          --positive integer
        MEMORY_PRIMITIVE => "auto",        --string; "auto", "distributed", "block" or "ultra" ;
        MEMORY_INIT_FILE => "none",        --string; "none" or "<filename>.mem"
        MEMORY_INIT_PARAM => "",          --string;
        USE_MEM_INIT     => 1,             --integer; 0,1
        WAKEUP_TIME      => "disable_sleep",--string; "disable_sleep" or "use_sleep_pin"
        MESSAGE_CONTROL  => 0,            --integer; 0,1

        -- Port A module generics
        READ_DATA_WIDTH_A => 32,          --positive integer
        ADDR_WIDTH_A     => 6,           --positive integer
        READ_RESET_VALUE_A => "0",       --string
        READ_LATENCY_A   => 2,           --non-negative integer
    )
    port map (
        -- Common module ports
        sleep      => '0',

        -- Port A module ports
        clka       => clka,
        rsta       => rsta,
        ena        => ena,
        regcea     => regcea,
        addra      => addra,
        injectsbiterra => '0',        --do not change
        injectdbiterra => '0',        --do not change
        douta      => douta,
        sbiterra   => open,          --do not change
        dbiterra   => open          --do not change
    );

-- End of xpm_memory_sprom_inst instance declaration
    
```

## Verilog Instantiation Template

```
// <--Cut the following instance declaration and paste it into
// the architecture statement part of the design-->

// xpm_memory_sprom: Single Port ROM
// Xilinx Parameterized Macro, Version 2016.1
xpm_memory_sprom # (

    // Common module parameters
    .MEMORY_SIZE      (2048),           //positive integer
    .MEMORY_PRIMITIVE ("auto"),        //string; "auto", "distributed", "block" or "ultra";
    .MEMORY_INIT_FILE ("none"),        //string; "none" or "<filename>.mem"
    .MEMORY_INIT_PARAM (""),           //string;
    .USE_MEM_INIT     (1),              //integer; 0,1
    .WAKEUP_TIME      ("disable_sleep"), //string; "disable_sleep" or "use_sleep_pin"
    .MESSAGE_CONTROL  (0),             //integer; 0,1

    // Port A module parameters
    .READ_DATA_WIDTH_A (32),           //positive integer
    .ADDR_WIDTH_A      (6),            //positive integer
    .READ_RESET_VALUE_A ("0"),         //string
    .READ_LATENCY_A    (2),            //non-negative integer

) xpm_memory_sprom_inst (

    // Common module ports
    .sleep      (1'b0),

    // Port A module ports
    .clka      (clka),
    .rsta      (rsta),
    .ena       (ena),
    .regcea    (regcea),
    .addra     (addra),
    .injectsbiterra (1'b0), //do not change
    .injectdbiterra (1'b0), //do not change
    .douta     (douta),
    .sbiterra  (), //do not change
    .dbiterra  (), //do not change

);

// End of xpm_memory_sprom instance declaration
```

## For More Information (individual items)

See the [UltraScale Programmable Devices Documentation](#).



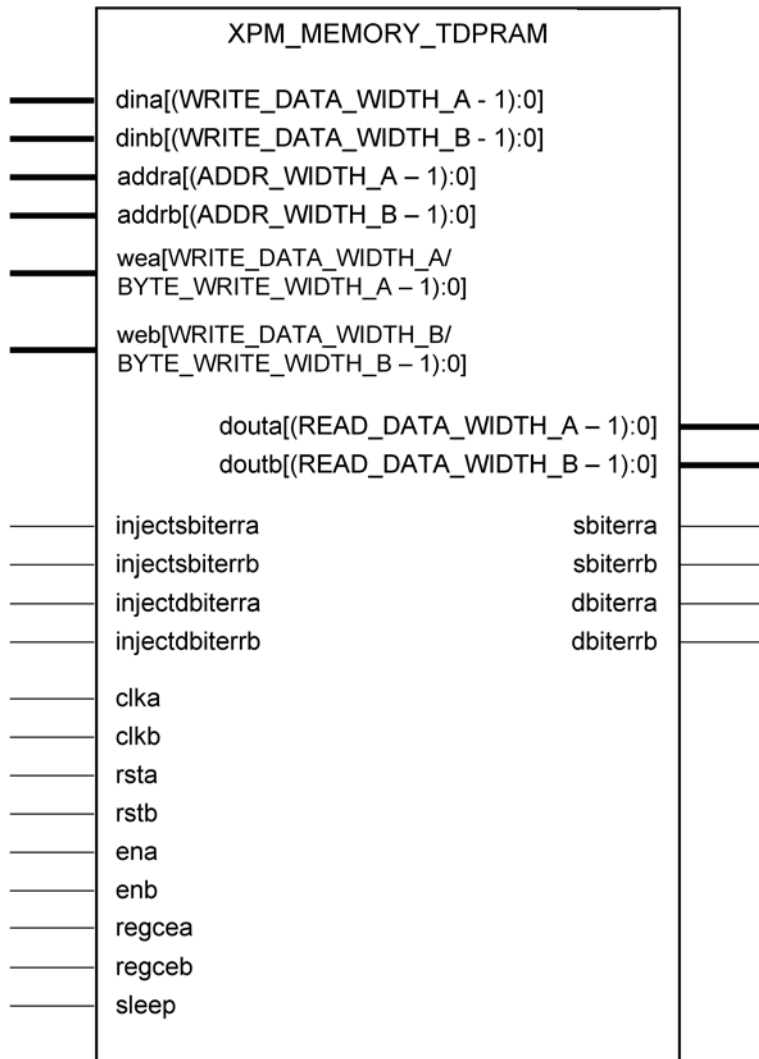
# XPM\_MEMORY\_TDPRAM

Parameterized Macro: True Dual Port RAM

MACRO\_GROUP: XPM

MACRO\_SUBGROUP: MEMORY

Families: UltraScale, UltraScale+



X16251-033016

## Introduction

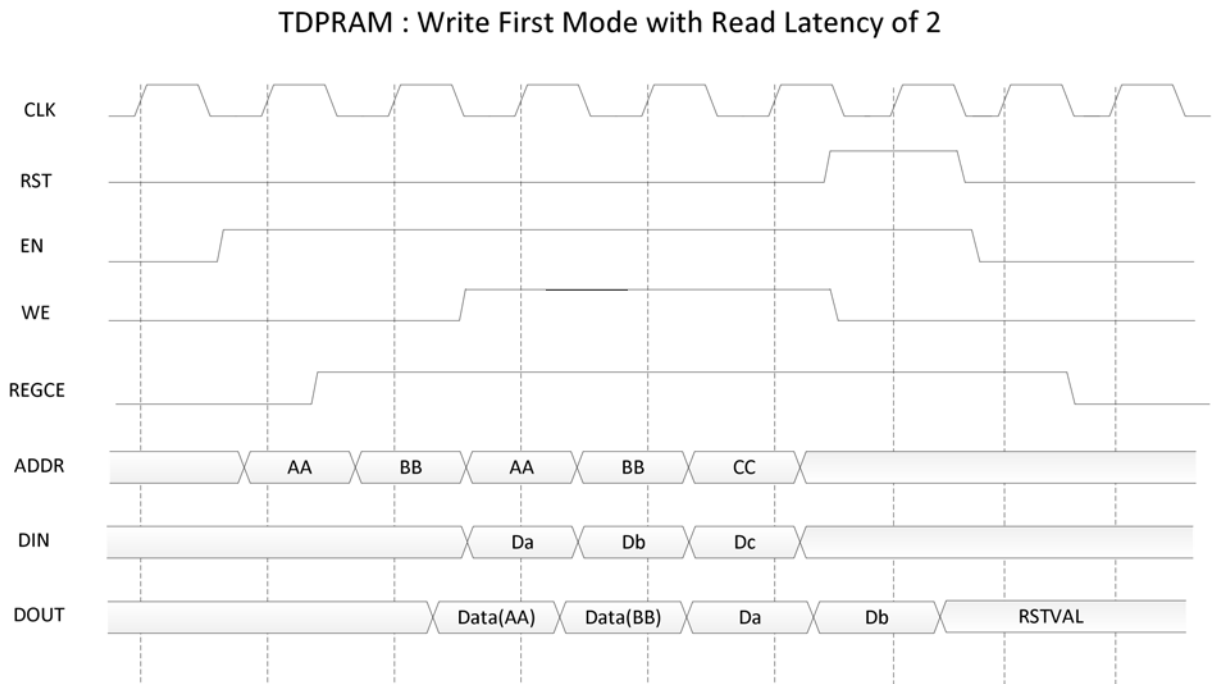
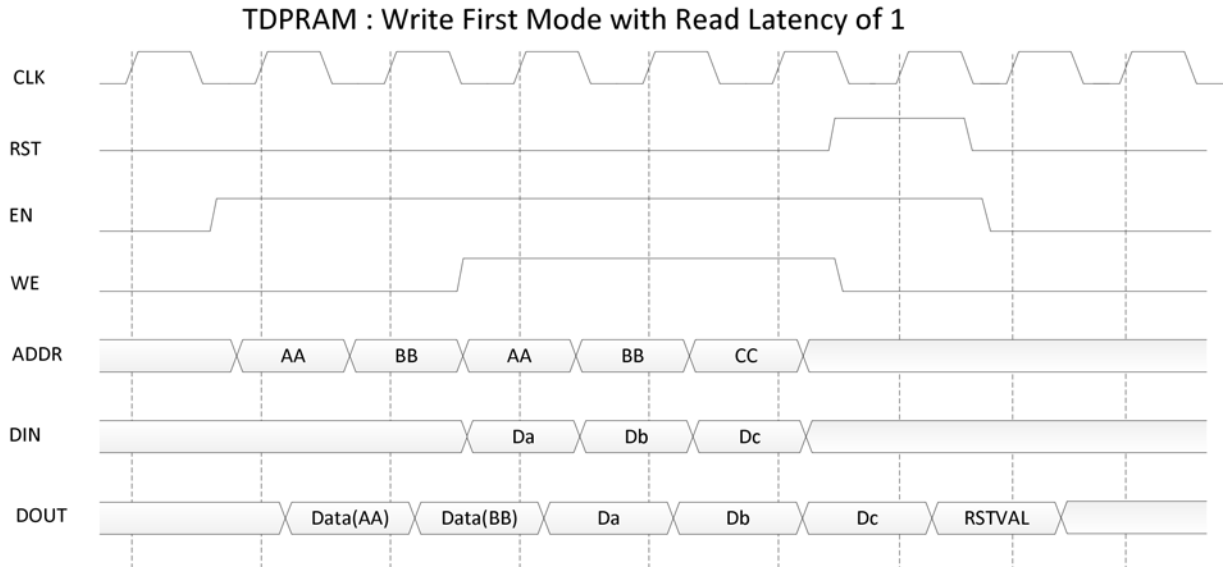
This Macro is used to Instantiate True Dual Port RAM. Read and writes to the memory can be done through Port-A and Port-B simultaneously.

The following describes the basic read and write port usage of an XPM\_MEMORY instance. It does not distinguish between ports A and B.

- All synchronous signals are sensitive to the rising edge of `clk[a|b]`, which is assumed to be a buffered and toggling clock signal behaving according to target device and memory primitive requirements.
- A read operation is implicitly performed to address `addr[a|b]` combinatorially. The data output is registered each `clk[a|b]` cycle that `en[a|b]` is asserted.
- Read data appears on the `dout[a|b]` port `READ_LATENCY_[A|B]` `clk[a|b]` cycles after the associated read operation.
- A write operation is explicitly performed, writing `din[a|b]` to address `addr[a|b]`, when both `en[a|b]` and `we[a|b]` are asserted on each `clk[a|b]` cycle.
- All read and write operations are gated by the value of `en[a|b]` on the initiating `clk[a|b]` cycle, regardless of input or output latencies. The `addr[a|b]` and `we[a|b]` inputs have no effect when `en[a|b]` is de-asserted on the coincident `clk[a|b]` cycle.
- The behavior of `dout[a|b]` with respect to the combination of `din[a|b]` and `addr[a|b]` is a function of `WRITE_MODE_[A|B]`.
- For each `clk[a|b]` cycle that `rst[a|b]` is asserted, the final output register is immediately but synchronously reset to `READ_RESET_VALUE_[A|B]`, irrespective of `READ_LATENCY_[A|B]`.
- For each `clk[a|b]` cycle that `regce[a|b]` is asserted and `rst[a|b]` is de-asserted, the final output register captures and outputs the value from the previous pipeline register.
- Undriven or unknown values provided on module inputs will produce undefined memory array and output port behavior.

**NOTE:** When the attribute "CLOCKING\_MODE" is set to "common\_clock", then all the read/write operations to the memory through port-A and port-B are performed on `clka`. If this attribute is set to "independent\_clock", then read/write operations through port-A are performed based on `clka`, and read/write operations through port-B are performed based on `clkb`.

## Timing Diagrams



Note : The above waveforms does not distinguish between port-A and Port-B, the behavior listed in the above waveforms is true for both Port-A and Port-B

## Port Description

Port	Direction	Width	Sense	Handling if unused	Description
CLKA	Input	1	Rising-edge	Required	Port A Clock
CLKB	Input	1	Rising-edge	Required	Port B Clock
RSTA	Input	1	NA	Required	Port A final output register reset to READ_RESET_VALUE_A
RSTB	Input	1	NA	Required	Port B final output register reset to READ_RESET_VALUE_B
ENA	Input	1	NA	Required	Port A Enable
ENB	Input	1	NA	Required	Port B Enable
REGCEA	Input	1	NA	Required	Port A final output pipeline register clock enable
REGCEB	Input	1	NA	Required	Port B final output pipeline register clock enable
WEA	Input	WRITE_DATA_WIDTH_A / BYTE_WRITE_WIDTH_A	NA	Required	Port A Write Enable
WEB	Input	WRITE_DATA_WIDTH_B / BYTE_WRITE_WIDTH_B	NA	Required	Port B Write Enable
ADDRA	Input	ADDR_WIDTH_A	NA	Required	Port A Address
ADDRB	Input	ADDR_WIDTH_B	NA	Required	Port B Address
DINA	Input	WRITE_DATA_WIDTH_A	NA	Required	Port A Write Data
DINB	Input	WRITE_DATA_WIDTH_B	NA	Required	Port B Write Data
INJECTSBITERRA	Input	1	NA	Tied to 0	Port A inject single bit error
INJECTSBITERRB	Input	1	NA	Tied to 0	Port B inject single bit error
INJECTDBITERRA	Input	1	NA	Tied to 0	Port A inject double bit error
INJECTDBITERRB	Input	1	NA	Tied to 0	Port B inject double bit error

Port	Direction	Width	Sense	Handling if unused	Description
SLEEP	Input	1	NA	Tied to 0	Memory sleep control. <ul style="list-style-type: none"> <li>Synchronous to clka when CLOCKING_MODE is "common_clock"</li> <li>Asynchronous to clk[a b] when CLOCKING_MODE is "independent_clock"</li> </ul> <b>NOTE:</b> Sleep is being sampled on both clka and clkb ports in the behavioral simulations. Make sure that sleep is asserted in such a way that it gets detected by both clka and clkb.
DOUTA	Output	READ_DATA_WIDTH_A	NA	Required	Port A output data
DOUTB	Output	READ_DATA_WIDTH_B	NA	Required	Port B output data
SBITERRA	Output	1	NA	Tied to 0	Port A single bit error indicator
SBITERRB	Output	1	NA	Tied to 0	Port B single bit error indicator
DBITERRA	Output	1	NA	Tied to 0	Port A double bit error indicator
DBITERRB	Output	1	NA	Tied to 0	Port B double bit error indicator

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
MEMORY_SIZE	Integer	>=1	2048	Total memory array size, in bits.
MEMORY_PRIMITIVE	string	"auto", "distributed", "block", "ultra"	"auto"	Memory primitive type to map to. <ul style="list-style-type: none"> <li>"auto" = allows synthesis to choose memory primitive</li> <li>"distributed" = Distributed Memory</li> <li>"block" = Block Memory</li> <li>"ultra" = UltraRAM Memory</li> </ul>
CLOCKING_MODE	string	"common_clock", "independent_clock"	"common_clock"	Clocking Mode.

Attribute	Type	Allowed Values	Default	Description
WRITE_DATA_WIDTH_[A B]	Integer	>=1	32	Width of the port [A B] write data port, din[a b].
BYTE_WRITE_WIDTH_[A B]	Integer	8, 9, WRITE_DATA_WIDTH_[A B]	WRITE_DATA_WIDTH_[A B]	Width of each byte, in bits, for enabled byte-wide write functionality of port [A B].
READ_DATA_WIDTH_[A B]	Integer	>=1	WRITE_DATA_WIDTH_[A]	Width of the port [A B] output data port, dout[a b].
READ_RESET_VALUE_[A B]	string	All type-legal values(hex string)	"0"	Initial and reset value of final register stage in port [A B] output data path.
READ_LATENCY_[A B]	Integer	>=1	2	Number of register stages in the port [A B] read data path.
WRITE_MODE_[A/B]	string	"write_first", "read_first", "no_change"	"no_change"	Write mode behavior for port [A/B] output data port, dout[a/b].
USE_MEMORY_INIT	Integer	0, 1	0	<ul style="list-style-type: none"> <li>1 = Enable the generation of below message</li> <li>0 = Disable the generation of below message</li> </ul> <p><b>NOTE:</b> This message gets generated only when there is no Memory Initialization specified either through file or parameter. INFO: MEMORY_INIT_FILE and MEMORY_INIT_PARAM together specifies no memory initialization. Initial memory contents will be all 0's</p>
MEMORY_INIT_FILE	Unspecified	"none", "<file_name>"	"none"	File name for memory content initialization.

Attribute	Type	Allowed Values	Default	Description
MEMORY_INIT_PARAM	Unspecified	"", valid hex String	""	<p>Initialize the contents of the memory using a parameter.</p> <p>Specify "" or "0" (including quotes) for no memory initialization through a parameter, or specify the string containing the hex characters. Each location must be separated by a delimiter (.).</p> <p>The parameter format must be ASCII and consist of only hexadecimal values organized into the specified depth by the narrowest data width generic value of the memory. For example, if the narrowest data width is 8, and the depth of memory is 8 locations, then the parameter value should be passed as shown below (0xAB in the below string is the 0th location and 0x78 is the 7th location).</p> <pre>MEMORY_INIT_PARAM = "AB,CD,EF,1,2,34,56,78"</pre> <p>Maximum supported memory size when MEMORY_INIT_PARAM is used is 4k bits.</p>
WAKEUP_TIME	Integer	"disable_sleep", "use_sleep_pin"	"disable_sleep"	Number of clk[a] cycles to wake, following de-assertion of sleep. "use_sleep_pin" has wake up time of 2 clock cycles.
VERSION	Integer	All type-legal values	0	Module version number.
MESSAGE_CONTROL	Integer	0, 1	1	Module messaging control. <ul style="list-style-type: none"> <li>0 = Disable dynamic messages (assertions)</li> <li>1 = Enable dynamic messages (assertions)</li> </ul>

## VHDL Instantiation Template

```
-- Copy the following two statements and paste them before the
-- entity declaration, unless they already exist.

Library xpm;
use xpm.vcomponents.all;

-- <--Cut the following instance declaration and paste it into
-- the architecture statement part of the design-->

-- xpm_memory_tdpram: True Dual Port RAM
```

```

-- Xilinx Parameterized Macro, Version 2016.1
xpm_memory_tdpram_inst : xpm_memory_tdpram
generic map (

    -- Common module generics
    MEMORY_SIZE           => 2048,           --positive integer
    MEMORY_PRIMITIVE     => "auto",         --string; "auto", "distributed", "block" or "ultra" ;
    CLOCKING_MODE        => "common_clock",--string; "common_clock", "independent_clock"
    MEMORY_INIT_FILE     => "none",         --string; "none" or "<filename>.mem"
    MEMORY_INIT_PARAM    => "",             --string;
    USE_MEM_INIT         => 1,              --integer; 0,1
    WAKEUP_TIME          => "disable_sleep",--string; "disable_sleep" or "use_sleep_pin"
    MESSAGE_CONTROL      => 0,              --integer;

    -- Port A module generics
    WRITE_DATA_WIDTH_A   => 32,             --positive integer
    READ_DATA_WIDTH_A   => 32,             --positive integer
    BYTE_WRITE_WIDTH_A  => 32,             --integer; 8, 9, or WRITE_DATA_WIDTH_A value
    ADDR_WIDTH_A        => 6,              --positive integer
    READ_RESET_VALUE_A  => "0",            --string
    READ_LATENCY_A      => 2,              --non-negative integer
    WRITE_MODE_A        => "no_change",    --string; "write_first", "read_first", "no_change"

    -- Port B module generics
    WRITE_DATA_WIDTH_B  => 32,             --positive integer
    READ_DATA_WIDTH_B   => 32,             --positive integer
    BYTE_WRITE_WIDTH_B  => 32,             --integer; 8, 9, or WRITE_DATA_WIDTH_B value
    ADDR_WIDTH_B        => 6,              --positive integer
    READ_RESET_VALUE_B  => "0",            --string
    READ_LATENCY_B      => 2,              --non-negative integer
    WRITE_MODE_B        => "no_change"    --string; "write_first", "read_first", "no_change"
)
port map (

    -- Common module ports
    sleep                => '0',

    -- Port A module ports
    clka                 => clka,
    rsta                 => rsta,
    ena                  => ena,
    regcea               => regcea,
    wea                  => wea,
    addra                => addra,
    dina                => dina,
    injectsbiterra      => '0',          --do not change
    injectdbiterra      => '0',          --do not change
    douta                => douta,
    sbiterra             => open,        --do not change
    dbiterra             => open,        --do not change

    -- Port B module ports
    clkb                 => clkb,
    rstb                 => rstb,
    enb                  => enb,
    regceb               => regceb,
    web                  => web,
    addrb                => addrb,
    dinb                 => dinb,
    injectsbiterrb      => '0',          --do not change
    injectdbiterrb     => '0',          --do not change
    doutb                => doutb,
    sbiterrb             => open,        --do not change
    dbiterrb             => open,        --do not change
);

-- End of xpm_memory_tdpram_inst instance declaration

```

## Verilog Instantiation Template



```
// <--Cut the following instance declaration and paste it into
// the architecture statement part of the design-->

// xpm_memory_tdpram: True Dual Port RAM
// Xilinx Parameterized Macro, Version 2016.1
xpm_memory_tdpram # (

    // Common module parameters
    .MEMORY_SIZE      (2048),           //positive integer
    .MEMORY_PRIMITIVE ("auto"),        //string; "auto", "distributed", "block" or "ultra";
    .CLOCKING_MODE    ("common_clock"), //string; "common_clock", "independent_clock"
    .MEMORY_INIT_FILE ("none"),        //string; "none" or "<filename>.mem"
    .MEMORY_INIT_PARAM (""),           //string;
    .USE_MEM_INIT     (1),             //integer; 0,1
    .WAKEUP_TIME      ("disable_sleep"), //string; "disable_sleep" or "use_sleep_pin"
    .MESSAGE_CONTROL  (0),             //integer; 0,1

    // Port A module parameters
    .WRITE_DATA_WIDTH_A (32),          //positive integer
    .READ_DATA_WIDTH_A  (32),          //positive integer
    .BYTE_WRITE_WIDTH_A (32),          //integer; 8, 9, or WRITE_DATA_WIDTH_A value
    .ADDR_WIDTH_A       (6),           //positive integer
    .READ_RESET_VALUE_A ("0"),         //string
    .READ_LATENCY_A     (2),           //non-negative integer
    .WRITE_MODE_A       ("no_change"), //string; "write_first", "read_first", "no_change"

    // Port B module parameters
    .WRITE_DATA_WIDTH_B (32),          //positive integer
    .READ_DATA_WIDTH_B  (32),          //positive integer
    .BYTE_WRITE_WIDTH_B (32),          //integer; 8, 9, or WRITE_DATA_WIDTH_B value
    .ADDR_WIDTH_B       (6),           //positive integer
    .READ_RESET_VALUE_B ("0"),         //vector of READ_DATA_WIDTH_B bits
    .READ_LATENCY_B     (2),           //non-negative integer
    .WRITE_MODE_B       ("no_change") //string; "write_first", "read_first", "no_change"

) xpm_memory_tdpram_inst (

    // Common module ports
    .sleep      (1'b0),

    // Port A module ports
    .clka       (clka),
    .rsta       (rsta),
    .ena        (ena),
    .regcea     (regcea),
    .wea        (wea),
    .addra      (addra),
    .dina       (dina),
    .injectsbiterra (1'b0), //do not change
    .injectdbiterra (1'b0), //do not change
    .douta      (douta),
    .sbiterra   (), //do not change
    .dbiterra   (), //do not change

    // Port B module ports
    .clkb       (clkb),
    .rstb       (rstb),
    .enb        (enb),
    .regceb     (regceb),
    .web        (web),
    .addrb      (addrb),
    .dinb       (dinb),
    .injectsbiterrb (1'b0), //do not change
    .injectdbiterrb (1'b0), //do not change
    .doutb      (doutb),
    .sbiterrb   (), //do not change
    .dbiterrb   (), //do not change

);

// End of xpm_memory_tdpram instance declaration
```

## For More Information (individual items)

See the [UltraScale Programmable Devices Documentation](#).

## Primitive Groups

The following Primitive Groups correlate to the PRIMITIVE\_GROUP cell property in the Vivado software. Similarly, the listed Primitive Subgroup correlates to the PRIMITIVE\_SUBGROUP property on the cells in the software. These can be used in filters in order to specify a class of cells for constraint processing and other tasks within Vivado.

ADVANCED	CLB	I/O
ARITHMETIC	CLOCK	REGISTER
BLOCKRAM	CONFIGURATION	

### ADVANCED

Design Element	Description	Primitive Subgroup
CMAC	Primitive: 100G MAC Block	MAC
CMACE4	Primitive: 100G MAC Block	MAC
GTHE3_CHANNEL	Primitive: Gigabit Transceiver for UltraScale devices	GT
GTHE3_COMMON	Primitive: Gigabit Transceiver for UltraScale devices	GT
GTHE4_CHANNEL	Primitive: Gigabit Transceiver for UltraScale+ devices	GT
GTHE4_COMMON	Primitive: Gigabit Transceiver for UltraScale+ devices	GT
GTYE3_CHANNEL	Primitive: Gigabit Transceiver for UltraScale devices	GT
GTYE3_COMMON	Primitive: Gigabit Transceiver for UltraScale devices	GT
GTYE4_CHANNEL	Primitive: Gigabit Transceiver for UltraScale+ devices	GT
GTYE4_COMMON	Primitive: Gigabit Transceiver for UltraScale+ devices	GT
IBUFDS_GTE3	Primitive: Gigabit Transceiver Buffer	GT
IBUFDS_GTE4	Primitive: Gigabit Transceiver Buffer	GT
ILKN	Primitive: Interlaken MAC	INTERLAKEN
ILKNE4	Primitive: Interlaken MAC	INTERLAKEN
OBUFDS_GTE3	Primitive: Gigabit Transceiver Buffer	GT
OBUFDS_GTE3_ADV	Primitive: Gigabit Transceiver Buffer	GT
OBUFDS_GTE4	Primitive: Gigabit Transceiver Buffer	GT
OBUFDS_GTE4_ADV	Primitive: Gigabit Transceiver Buffer	GT
PCIE40E4	Primitive: Integrated Block for PCI Express	PCIE

Design Element	Description	Primitive Subgroup
<a href="#">PCIE_3_1</a>	Primitive: Integrated Block for PCI Express	PCIE
<a href="#">SYSMONE1</a>	Primitive: Xilinx Analog-to-Digital Converter and System Monitor	SYSMON
<a href="#">SYSMONE4</a>	Primitive: Xilinx Analog-to-Digital Converter and System Monitor	SYSMON

## ARITHMETIC

Design Element	Description	Primitive Subgroup
<a href="#">DSP48E2</a>	Primitive: 48-bit Multi-Functional Arithmetic Block	DSP

## BLOCKRAM

Design Element	Description	Primitive Subgroup
<a href="#">FIFO18E2</a>	Primitive: 18Kb FIFO (First-In-First-Out) Block RAM Memory	FIFO
<a href="#">FIFO36E2</a>	Primitive: 36Kb FIFO (First-In-First-Out) Block RAM Memory	FIFO
<a href="#">RAMB18E2</a>	Primitive: 18K-bit Configurable Synchronous Block RAM	BRAM
<a href="#">RAMB36E2</a>	Primitive: 36K-bit Configurable Synchronous Block RAM	BRAM
<a href="#">URAM288</a>	Primitive: 288K-bit High-Density Memory Building Block	URAM
<a href="#">URAM288_BASE</a>	Primitive: 288K-bit High-Density Base Memory Building Block	URAM

## CLB

Design Element	Description	Primitive Subgroup
<a href="#">LUT6_2</a>	Primitive: Six-input, 2-output, Look-Up Table	LUT
<a href="#">RAM128X1D</a>	Primitive: 128-Deep by 1-Wide Dual Port Random Access Memory (Select RAM)	LUTRAM
<a href="#">RAM128X1S</a>	Primitive: 128-Deep by 1-Wide Random Access Memory (Select RAM)	LUTRAM
<a href="#">RAM256X1D</a>	Primitive: 256-Deep by 1-Wide Dual Port Random Access Memory (Select RAM)	LUTRAM
<a href="#">RAM256X1S</a>	Primitive: 256-Deep by 1-Wide Random Access Memory (Select RAM)	LUTRAM

Design Element	Description	Primitive Subgroup
RAM32M	Primitive: 32-Deep by 8-bit Wide Multi Port Random Access Memory (Select RAM)	LUTRAM
RAM32M16	Primitive: 32-Deep by 16-bit Wide Multi Port Random Access Memory (Select RAM)	LUTRAM
RAM32X1D	Primitive: 32-Deep by 1-Wide Static Dual Port Synchronous RAM	LUTRAM
RAM32X1S	Primitive: 32-Deep by 1-Wide Static Synchronous RAM	LUTRAM
RAM512X1S	Primitive: 512-Deep by 1-Wide Random Access Memory (Select RAM)	LUTRAM
RAM64M	Primitive: 64-Deep by 4-bit Wide Multi Port Random Access Memory (Select RAM)	LUTRAM
RAM64M8	Primitive: 64-Deep by 8-bit Wide Multi Port Random Access Memory (Select RAM)	LUTRAM
RAM64X1D	Primitive: 64-Deep by 1-Wide Dual Port Static Synchronous RAM	LUTRAM
RAM64X1S	Primitive: 64-Deep by 1-Wide Static Synchronous RAM	LUTRAM
AND2B1L	Primitive: Two input AND gate implemented in place of a CLB Latch	LATCH
CARRY8	Primitive: Fast Carry Logic with Look Ahead	CARRY
CFGLUT5	Primitive: 5-input Dynamically Reconfigurable Look-Up Table (LUT)	LUT
LUT1	Primitive: 1-Bit Look-Up Table	LUT
LUT2	Primitive: 2-Bit Look-Up Table	LUT
LUT3	Primitive: 3-Bit Look-Up Table	LUT
LUT4	Primitive: 4-Bit Look-Up Table	LUT
LUT5	Primitive: 5-Bit Look-Up Table	LUT
LUT6	Primitive: 6-Bit Look-Up Table	LUT
MUXF7	Primitive: CLB MUX to connect two LUT6's Together	MUXF
MUXF8	Primitive: CLB MUX to connect two MUXF7's Together	MUXF
MUXF9	Primitive: CLB MUX to connect two MUXF8's Together	MUXF
OR2L	Primitive: Two input OR gate implemented in place of a CLB Latch	LATCH
RAM64X8SW	Primitive: 64-Deep by 8-bit Wide Random Access Memory with Single-Bit Write (Select RAM)	LUTRAM
SRL16E	Primitive: 16-Bit Shift Register Look-Up Table (LUT)	SRL
SRLC32E	Primitive: 32-Bit Shift Register Look-Up Table (LUT)	SRL

## CLOCK

Design Element	Description	Primitive Subgroup
BUFG	Primitive: General Clock Buffer	BUFFER
BUFG_GT	Primitive: Clock Buffer Driven by Gigabit Transceiver	BUFFER
BUFG_GT_SYNC	Primitive: Synchronizer for BUFG_GT Control Signals	CLOCK_SYNC
BUFG_PS	Primitive: A high-fanout buffer for low-skew distribution of the PS Clock signals	BUFFER
BUFGCE	Primitive: General Clock Buffer with Clock Enable	BUFFER
BUFGCE_1	Primitive: General Clock Buffer with Clock Enable and Output State 1	BUFFER
BUFGCE_DIV	Primitive: General Clock Buffer with Divide Function	BUFFER
BUFGCTRL	Primitive: General Clock Control Buffer	MUX
BUFGMUX	Primitive: General Clock Mux Buffer	MUX
BUFGMUX_1	Primitive: General Clock Mux Buffer with Output State 1	MUX
BUFGMUX_CTRL	Primitive: 2-to-1 General Clock MUX Buffer	MUX
MMCME3_ADV	Primitive: Advanced Mixed Mode Clock Manager (MMCM)	PLL
MMCME3_BASE	Primitive: Base Mixed Mode Clock Manager (MMCM)	PLL
MMCME4_ADV	Primitive: Advanced Mixed Mode Clock Manager (MMCM)	PLL
PLLE3_ADV	Primitive: Advanced Phase-Locked Loop (PLL)	PLL
PLLE3_BASE	Primitive: Base Phase-Locked Loop (PLL)	PLL
PLLE4_ADV	Primitive: Advanced Phase-Locked Loop (PLL)	PLL

## CONFIGURATION

Design Element	Description	Primitive Subgroup
BSCANE2	Primitive: Boundary-Scan User Instruction	BSCAN
DNA_PORTE2	Primitive: Device DNA Access Port	DNA
EFUSE_USR	Primitive: 32-bit non-volatile design ID	EFUSE
FRAME_ECCE3	Primitive: Configuration Frame Error Correction	ECC
ICAPE3	Primitive: Internal Configuration Access Port	ICAP
MASTER_JTAG	Primitive: JTAG Port Access	MASTER_JTAG
STARTUPE3	Primitive: STARTUP Block	STARTUP
USR_ACCESSE2	Primitive: Configuration Data Access	USR_ACCESS

## I/O

Design Element	Description	Primitive Subgroup
<a href="#">BITSlice_CONTROL</a>	Primitive: BITSlice_CONTROL for control using Native Mode	BITSlice
<a href="#">DCIRESET</a>	Primitive: Digitally Controlled Impedance Reset Component	DCI_RESET
<a href="#">HPIO_VREF</a>	Primitive: VREF Scan	INPUT_BUFFER
<a href="#">IBUF</a>	Primitive: Input Buffer	INPUT_BUFFER
<a href="#">IBUF_ANALOG</a>	Primitive: Analog Auxiliary SYSMON Input Buffer	INPUT_BUFFER
<a href="#">IBUF_IBUFDISABLE</a>	Primitive: Input Buffer With Input Buffer Disable	INPUT_BUFFER
<a href="#">IBUF_INTERMDISABLE</a>	Primitive: Input Buffer With Input Buffer Disable and On-die Input Termination Disable	INPUT_BUFFER
<a href="#">IBUFDS</a>	Primitive: Differential Input Buffer	INPUT_BUFFER
<a href="#">IBUFDS_DIFF_OUT</a>	Primitive: Differential Input Buffer With Complementary Outputs	INPUT_BUFFER
<a href="#">IBUFDS_DIFF_OUT_IBUFDISABLE</a>	Primitive: Differential Input Buffer With Complementary Outputs and Input Buffer Disable	INPUT_BUFFER
<a href="#">IBUFDS_DIFF_OUT_INTERMDISABLE</a>	Primitive: Differential Input Buffer with Complementary Outputs, Input Path Disable and On-die Input Termination Disable	INPUT_BUFFER
<a href="#">IBUFDS_DPHY</a>	Primitive: Differential Input Buffer with MIPI support	INPUT_BUFFER
<a href="#">IBUFDS_IBUFDISABLE</a>	Primitive: Differential Input Buffer With Input Buffer Disable	INPUT_BUFFER
<a href="#">IBUFDS_INTERMDISABLE</a>	Primitive: Differential Input Buffer With Input Buffer Disable and On-die Input Termination Disable	INPUT_BUFFER
<a href="#">IBUFDSE3</a>	Primitive: Differential Input Buffer with Offset Calibration	INPUT_BUFFER
<a href="#">IBUFE3</a>	Primitive: Input Buffer with Offset Calibration and VREF Tuning	INPUT_BUFFER
<a href="#">IDELAYCTRL</a>	Primitive: IDELAYE3/ODELAYE3 Tap Delay Value Control	DELAY
<a href="#">IDELAYE3</a>	Primitive: Input Fixed or Variable Delay Element	DELAY
<a href="#">IOBUF</a>	Primitive: Input/Output Buffer	BIDIR_BUFFER
<a href="#">IOBUF_DCIE</a>	Primitive: Input/Output Buffer DCI Enable	BIDIR_BUFFER
<a href="#">IOBUF_INTERMDISABLE</a>	Primitive: Bidirectional Buffer with Input Path Disable and On-die Input Termination Disable	BIDIR_BUFFER
<a href="#">IOBUFDS</a>	Primitive: Differential Input/Output Buffer	BIDIR_BUFFER
<a href="#">IOBUFDS_DCIE</a>	Primitive: Differential Bidirectional Buffer With Input Buffer Disable and On-die Input Termination Disable	BIDIR_BUFFER
<a href="#">IOBUFDS_DIFF_OUT</a>	Primitive: Differential Input/Output Buffer Primitive With Complementary Outputs for the Input Buffer	BIDIR_BUFFER

Design Element	Description	Primitive Subgroup
IOBUFDS_DIFF_OUT_DCEN	Primitive: Differential Bidirectional Buffer with Complementary Outputs, Input Path Disable, and On-die Input Termination Disable	BIDIR_BUFFER
IOBUFDS_DIFF_OUT_INTERMDISABLE	Primitive: Differential Bidirectional Buffer with Complementary Outputs, Input Buffer Disable and On-die Input Termination Disable	BIDIR_BUFFER
IOBUFDS_INTERMDISABLE	Primitive: Differential Bidirectional Buffer With Input Buffer Disable and On-die Input	BIDIR_BUFFER
IOBUFDSE3	Primitive: Differential Bidirectional I/O Buffer with Offset Calibration	BIDIR_BUFFER
IOBUFE3	Primitive: Bidirectional I/O Buffer with Offset Calibration and VREF Tuning	BIDIR_BUFFER
ISERDESE3	Primitive: Input SERIAL/DESerializer	SERDES
KEEPER	Primitive: I/O Weak Keeper	WEAK_DRIVER
OBUF	Primitive: Output Buffer	OUTPUT_BUFFER
OBUFDS	Primitive: Differential Output Buffer	OUTPUT_BUFFER
OBUFDS_DPHY	Primitive: Differential Output Buffer with MIPI support	OUTPUT_BUFFER
OBUFT	Primitive: 3-State Output Buffer	OUTPUT_BUFFER
OBUFTDS	Primitive: Differential 3-state Output Buffer	OUTPUT_BUFFER
ODELAYE3	Primitive: Output Fixed or Variable Delay Element	DELAY
OSERDESE3	Primitive: Output SERIAL/DESerializer	SERDES
PULLDOWN	Primitive: I/O Pulldown	WEAK_DRIVER
PULLUP	Primitive: I/O Pullup	WEAK_DRIVER
RIU_OR	Primitive: Register Interface Unit Selection Block	BITSLICE
RX_BITSLICE	Primitive: RX_BITSLICE for input using Native Mode	BITSLICE
RXTX_BITSLICE	Primitive: RXTX_BITSLICE for bidirectional I/O using Native Mode	BITSLICE
TX_BITSLICE	Primitive: TX_BITSLICE for output using Native Mode	BITSLICE
TX_BITSLICE_TRI	Primitive: TX_BITSLICE_TRI for tristate using Native Mode	BITSLICE

## REGISTER

Design Element	Description	Primitive Subgroup
FDCE	Primitive: D Flip-Flop with Clock Enable and Asynchronous Clear	SDR
FDPE	Primitive: D Flip-Flop with Clock Enable and Asynchronous Preset	SDR



Design Element	Description	Primitive Subgroup
FDRE	Primitive: D Flip-Flop with Clock Enable and Synchronous Reset	SDR
FDSE	Primitive: D Flip-Flop with Clock Enable and Synchronous Set	SDR
HARD_SYNC	Primitive: Metastability Hardened Registers	METASTABILITY
IDDRE1	Primitive: Dedicated Dual Data Rate (DDR) Input Register	DDR
LDCE	Primitive: Transparent Latch with Clock Enable and Asynchronous Clear	LATCH
LDPE	Primitive: Transparent Latch with Clock Enable and Asynchronous Preset	LATCH
ODDRE1	Primitive: Dedicated Dual Data Rate (DDR) Output Register	DDR

## Design Elements

---

### About Design Elements

This section describes the design elements that can be used with UltraScale™ architecture-based 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

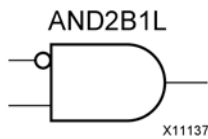
## AND2B1L

Primitive: Two input AND gate implemented in place of a CLB Latch

PRIMITIVE\_GROUP: CLB

PRIMITIVE\_SUBGROUP: LATCH

Families: UltraScale, UltraScale+



### Introduction

This element allows the specification of a configurable CLB latch to take the function of a two input AND gate. The use of this element can reduce logic levels and increase logic density of the part by trading off register/latch resources for logic. Xilinx suggests caution when using this component as it can affect register packing and density within a CLB.

### Logic Table

Inputs		Outputs
DI	SRI	O
0	0	0
0	1	0
1	0	1
1	1	0

### Port Descriptions

Port	Direction	Width	Function
DI	Input	1	Active-High input that is generally connected to sourcing LUT located in the same CLB.
O	Output	1	Output of the AND gate.
SRI	Input	1	Input that is generally source from outside of the CLB. The attribute IS_SRI_INVERTED determines the active polarity of this signal.  <b>NOTE:</b> To allow more than one AND2B1L or OR2L to be packed into a half CLB, a common signal must be connected to this input.

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
IS_SRI_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion for the SRI pin of this component. When set to high, the AND2B1L acts as a true AND gate.

## VHDL Instantiation Template

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- AND2B1L: Two input AND gate implemented in place of a CLB Latch
--           UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

AND2B1L_inst : AND2B1L
generic map (
    IS_SRI_INVERTED => '0' -- Optional inversion for SRI
)
port map (
    O => O,      -- 1-bit output: AND gate output
    DI => DI,    -- 1-bit input: Data input connected to LUT logic
    SRI => SRI   -- 1-bit input: External CLB data
);

-- End of AND2B1L_inst instantiation
    
```

## Verilog Instantiation Template

```

// AND2B1L: Two input AND gate implemented in place of a CLB Latch
//           UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

AND2B1L #(
    .IS_SRI_INVERTED(1'b0) // Optional inversion for SRI
)
AND2B1L_inst (
    .O(O),      // 1-bit output: AND gate output
    .DI(DI),    // 1-bit input: Data input connected to LUT logic
    .SRI(SRI)  // 1-bit input: External CLB data
);

// End of AND2B1L_inst instantiation
    
```

## For More Information

- See the *UltraScale Architecture Configurable Logic Block User Guide* ([UG574](#)).
- See the [UltraScale Programmable Devices Documentation](#).

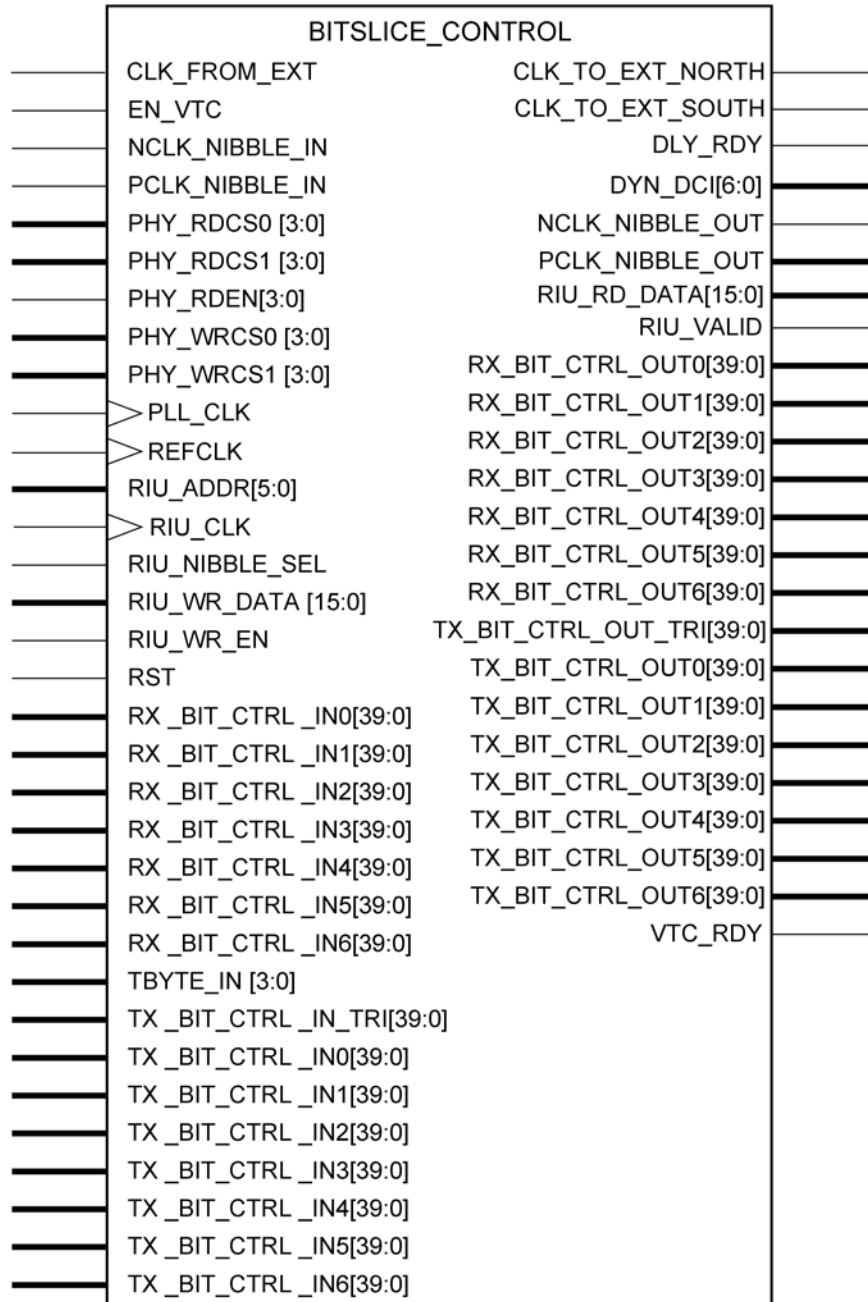
# BITSLICE\_CONTROL

Primitive: BITSLICE\_CONTROL for control using Native Mode

PRIMITIVE\_GROUP: I/O

PRIMITIVE\_SUBGROUP: BITSLICE

Families: UltraScale, UltraScale+



X13411-102815

## Introduction

In native mode, the BITSlice\_CONTROL controls the clocking and characteristics of the six or seven bitslices within a nibble.

## Port Descriptions

Port	Direction	Width	Function
CLK_FROM_EXT	Input	1	Inter-byte clock coming from north or south BITSlice_CONTROL
CLK_TO_EXT_NORTH	Output	1	Inter-byte clock going to north BITSlice_CONTROL
CLK_TO_EXT_SOUTH	Output	1	Inter-byte clock going to south BITSlice_CONTROL
DLY_RDY	Output	1	Fixed delay calibration complete
DYN_DCI<6:0>	Output	7	Direct control of IOB DCI when using a memory interface
EN_VTC	Input	1	Enables voltage and temperature compensation when High
NCLK_NIBBLE_IN	Input	1	Intra-byte DQS strobes from other/clock control block
NCLK_NIBBLE_OUT	Output	1	Intra-byte DQS strobes/clock to other control block
PCLK_NIBBLE_IN	Input	1	Intra-byte DQS strobes/clock from other control block
PCLK_NIBBLE_OUT	Output	1	Intra-byte DQS strobes/clock to other control block
PHY_RDCS0<3:0>	Input	4	Rank select. Selects one of four ranks when using a memory interface.
PHY_RDCS1<3:0>	Input	4	Rank select. Selects one of four ranks when using a memory interface.
PHY_RDEN<3:0>	Input	4	Read burst enable when using a memory interface
PHY_WRCS0<3:0>	Input	4	Rank select. Selects one of four ranks when using a memory interface.
PHY_WRCS1<3:0>	Input	4	Rank select. Selects one of four ranks when using a memory interface.
PLL_CLK	Input	1	PLL clock input
REFCLK	Input	1	Frequency reference clock for delay control
RIU_ADDR<5:0>	Input	6	Address input for RIU
RIU_CLK	Input	1	System clock from fabric for RIU access
RIU_NIBBLE_SEL	Input	1	Nibble select to enable RIU read/write
RIU_RD_DATA<15:0>	Output	16	RIU Output Read data to the controller
RIU_VALID	Output	1	Indicates that last data written has been accepted when High
RIU_WR_DATA<15:0>	Input	16	RIU Input Write data from the controller
RIU_WR_EN	Input	1	Enables write to RIU when High
RST	Input	1	Asynchronous global reset
RX_BIT_CTRL_IN0<39:0>	Input	40	Input control and data bus from Bitslice 0
RX_BIT_CTRL_IN1<39:0>	Input	40	Input control and data bus from Bitslice 1

Port	Direction	Width	Function
RX_BIT_CTRL_IN2<39:0>	Input	40	Input control and data bus from Bitslice 2
RX_BIT_CTRL_IN3<39:0>	Input	40	Input control and data bus from Bitslice 3
RX_BIT_CTRL_IN4<39:0>	Input	40	Input control and data bus from Bitslice 4
RX_BIT_CTRL_IN5<39:0>	Input	40	Input control and data bus from Bitslice 5
RX_BIT_CTRL_IN6<39:0>	Input	40	Input control and data bus from Bitslice 6
RX_BIT_CTRL_OUT0<39:0>	Output	40	Output control and data bus to Bitslice 0
RX_BIT_CTRL_OUT1<39:0>	Output	40	Output control and data bus to Bitslice 1
RX_BIT_CTRL_OUT2<39:0>	Output	40	Output control and data bus to Bitslice 2
RX_BIT_CTRL_OUT3<39:0>	Output	40	Output control and data bus to Bitslice 3
RX_BIT_CTRL_OUT4<39:0>	Output	40	Output control and data bus to Bitslice 4
RX_BIT_CTRL_OUT5<39:0>	Output	40	Output control and data bus to Bitslice 5
RX_BIT_CTRL_OUT6<39:0>	Output	40	Output control and data bus to Bitslice 6
TBYTE_IN<3:0>	Input	4	Output enable for 3-state control and WClkgen when using a memory interface
TX_BIT_CTRL_IN_TRI<39:0>	Input	40	Input control and data bus from 3-state TX_BITSLICE_TRI
TX_BIT_CTRL_IN0<39:0>	Input	40	Input control and data bus from Bitslice 0
TX_BIT_CTRL_IN1<39:0>	Input	40	Input control and data bus from Bitslice 1
TX_BIT_CTRL_IN2<39:0>	Input	40	Input control and data bus from Bitslice 2
TX_BIT_CTRL_IN3<39:0>	Input	40	Input control and data bus from Bitslice 3
TX_BIT_CTRL_IN4<39:0>	Input	40	Input control and data bus from Bitslice 4
TX_BIT_CTRL_IN5<39:0>	Input	40	Input control and data bus from Bitslice 5
TX_BIT_CTRL_IN6<39:0>	Input	40	Input control and data bus from Bitslice 6
TX_BIT_CTRL_OUT_TRI<39:0>	Output	40	Output control and data bus to 3-state TX_BITSLICE_TRI
TX_BIT_CTRL_OUT0<39:0>	Output	40	Output control and data bus to Bitslice 0
TX_BIT_CTRL_OUT1<39:0>	Output	40	Output control and data bus to Bitslice 1
TX_BIT_CTRL_OUT2<39:0>	Output	40	Output control and data bus to Bitslice 2
TX_BIT_CTRL_OUT3<39:0>	Output	40	Output control and data bus to Bitslice 3
TX_BIT_CTRL_OUT4<39:0>	Output	40	Output control and data bus to Bitslice 4
TX_BIT_CTRL_OUT5<39:0>	Output	40	Output control and data bus to Bitslice 5
TX_BIT_CTRL_OUT6<39:0>	Output	40	Output control and data bus to Bitslice 6
VTC_RDY	Output	1	PHY calibration is complete, VTC is enabled after EN_VTC is enabled



## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	Yes

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
DIV_MODE	STRING	"DIV2", "DIV4"	"DIV2"	Select between controller DIV2 or DIV4 mode
EN_CLK_TO_EXT_NORTH	STRING	"DISABLE", "ENABLE"	"DISABLE"	Enable clock forwarding to north for inter-byte clocking
EN_CLK_TO_EXT_SOUTH	STRING	"DISABLE", "ENABLE"	"DISABLE"	Enable clock forwarding to south for inter-byte clocking
EN_DYN_ODLY_MODE	STRING	"FALSE", "TRUE"	"FALSE"	Enable dynamic output delay mode when TRUE
EN_OTHER_NCLK	STRING	"FALSE", "TRUE"	"FALSE"	<ul style="list-style-type: none"> <li>"TRUE": Select the NCLK from the other BITSlice_CONTROL in the nibble</li> <li>"FALSE": Other BITSlice_CONTROL NCLK is not used</li> </ul>
EN_OTHER_PCLK	STRING	"FALSE", "TRUE"	"FALSE"	<ul style="list-style-type: none"> <li>"TRUE": Select the PCLK from the other BITSlice_CONTROL in the nibble</li> <li>"FALSE": Other BITSlice_CONTROL PCLK is not used</li> </ul>
IDLY_VT_TRACK	STRING	"TRUE", "FALSE"	"TRUE"	Globally enable VT tracking for input delays associated with the BITSlice_CONTROL
INV_RXCLK	STRING	"FALSE", "TRUE"	"FALSE"	Invert clock path from IOB to upper RX bitslice
ODLY_VT_TRACK	STRING	"TRUE", "FALSE"	"TRUE"	Globally enable VT tracking for output delays associated with the BITSlice_CONTROL
QDLY_VT_TRACK	STRING	"TRUE", "FALSE"	"TRUE"	Globally enable VT tracking for clock delays associated with the BITSlice_CONTROL
READ_IDLE_COUNT	HEX	6'h00 to 6'h3f	6'h00	Gap count between read bursts for ODT control counter
REFCLK_SRC	STRING	"PLLCLK", "REFCLK"	"PLLCLK"	Selects the PLLCLK or REFCLK as the input clock for the delay control

Attribute	Type	Allowed Values	Default	Description
ROUNDING_FACTOR	DECIMAL	16, 2, 4, 8, 32, 64, 128	16	Rounding factor in BISC spec
RX_CLK_PHASE_N	STRING	"SHIFT_0", "SHIFT_90"	"SHIFT_0"	<ul style="list-style-type: none"> <li>"SHIFT_0": No Shift</li> <li>"SHIFT_90": Shift Read CLK by 90 relative to read DQ during calibration</li> </ul>
RX_CLK_PHASE_P	STRING	"SHIFT_0", "SHIFT_90"	"SHIFT_0"	<ul style="list-style-type: none"> <li>"SHIFT_0": No Shift</li> <li>"SHIFT_90": Shift Read CLK by 90 relative to read DQ during calibration</li> </ul>
RX_GATING	STRING	"DISABLE", "ENABLE"	"DISABLE"	ENABLE/DISABLE read DQS gating
RXGATE_EXTEND	STRING	"FALSE", "TRUE"	"FALSE"	Reserved for use by MIG Memory Controller. Do Not Change.
SELF_CALIBRATE	STRING	"ENABLE", "DISABLE"	"ENABLE"	Enable or Disable Built in Self Calibration of the nibble group controlled by the BITSlice_CONTROL
SERIAL_MODE	STRING	"FALSE", "TRUE"	"FALSE"	Put BITSlice read paths into serial mode. The input clock from the data receiver comes from an external source via a PLLE3. One example use is for SGMII.
SIM_DEVICE	STRING	"ULTRASCALE", "ULTRASCALE_PLUS_ES1"	"ULTRASCALE"	Set the device version
TX_GATING	STRING	"DISABLE", "ENABLE"	"DISABLE"	ENABLE/DISABLE clock gating in WClkgen

## VHDL Instantiation Template

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- BITSlice_CONTROL: BITSlice_CONTROL for control using Native Mode
--                               UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

BITSlice_CONTROL_inst : BITSlice_CONTROL
generic map (
    DIV_MODE => "DIV2",           -- Controller DIV2/DIV4 mode (DIV2, DIV4)
    EN_CLK_TO_EXT_NORTH => "DISABLE", -- Enable clock forwarding to north
    EN_CLK_TO_EXT_SOUTH => "DISABLE", -- Enable clock forwarding to south
    EN_DYN_ODLY_MODE => "FALSE",   -- Enable dynamic output delay mode
    EN_OTHER_NCLK => "FALSE",      -- Select the NCLK from the other BITSlice_CONTROL in the nibble
    -- (FALSE, TRUE)
    EN_OTHER_PCLK => "FALSE",     -- Select the PCLK from the other BITSlice_CONTROL in the nibble
    -- (FALSE, TRUE)
    IDLY_VT_TRACK => "TRUE",      -- Enable VT tracking for input delays
    INV_RXCLK => "FALSE",         -- Invert clock path from IOB to upper RX bitslice

```

```

ODLY_VT_TRACK => "TRUE",           -- Enable VT tracking for output delays
QDLY_VT_TRACK => "TRUE",           -- Enable VT tracking for clock delays
READ_IDLE_COUNT => X"00",          -- Gap count between read bursts for ODT control counter (0-3f)
REFCLK_SRC => "PLLCLK",            -- Select the input clock for the delay control (PLLCLK, REFCLK)
ROUNDING_FACTOR => 16,              -- Rounding factor in BISC spec (128-8)
RXGATE_EXTEND => "FALSE",          -- Reserved for use by MIG Memory Controller. Do Not Change.
RX_CLK_PHASE_N => "SHIFT_0",      -- Shift the Read CLK relative to read DQ during calibration (SHIFT_0,
-- SHIFT_90)
RX_CLK_PHASE_P => "SHIFT_0",      -- Shift the Read CLK relative to read DQ during calibration (SHIFT_0,
-- SHIFT_90)
RX_GATING => "DISABLE",            -- ENABLE/DISABLE read DQS gating
SELF_CALIBRATE => "ENABLE",        -- Enable BISC of nibble controlled by BITSlice_CONTROL
SERIAL_MODE => "FALSE",           -- Put BITSlice read paths into serial mode (FALSE, TRUE)
SIM_DEVICE => "ULTRASCALE",       -- Set the device version (ULTRASCALE, ULTRASCALE_PLUS_ES1)
TX_GATING => "DISABLE"            -- ENABLE/DISABLE clock gating in Wclkgen
)
port map (
  CLK_TO_EXT_NORTH => CLK_TO_EXT_NORTH,    -- 1-bit output: Inter-byte clock going to north
-- BITSlice_CONTROL

  CLK_TO_EXT_SOUTH => CLK_TO_EXT_SOUTH,    -- 1-bit output: Inter-byte clock going to south
-- BITSlice_CONTROL

  DLY_RDY => DLY_RDY,                      -- 1-bit output: Fixed delay calibration complete
  DYN_DCI => DYN_DCI,                      -- 7-bit output: Direct control of IOB DCI when using a
-- memory interface

  NCLK_NIBBLE_OUT => NCLK_NIBBLE_OUT,      -- 1-bit output: Intra-byte DQS strobes/clock to other
-- control block

  PCLK_NIBBLE_OUT => PCLK_NIBBLE_OUT,      -- 1-bit output: Intra-byte DQS strobes/clock to other
-- control block

  RIU_RD_DATA => RIU_RD_DATA,              -- 16-bit output: RIU Output Read data to the controller
  RIU_VALID => RIU_VALID,                  -- 1-bit output: Last data written has been accepted when High

  RX_BIT_CTRL_OUT0 => RX_BIT_CTRL_OUT0,    -- 40-bit output: Output bus to Bitslice 0
  RX_BIT_CTRL_OUT1 => RX_BIT_CTRL_OUT1,    -- 40-bit output: Output bus to Bitslice 1
  RX_BIT_CTRL_OUT2 => RX_BIT_CTRL_OUT2,    -- 40-bit output: Output bus to Bitslice 2
  RX_BIT_CTRL_OUT3 => RX_BIT_CTRL_OUT3,    -- 40-bit output: Output bus to Bitslice 3
  RX_BIT_CTRL_OUT4 => RX_BIT_CTRL_OUT4,    -- 40-bit output: Output bus to Bitslice 4
  RX_BIT_CTRL_OUT5 => RX_BIT_CTRL_OUT5,    -- 40-bit output: Output bus to Bitslice 5
  RX_BIT_CTRL_OUT6 => RX_BIT_CTRL_OUT6,    -- 40-bit output: Output bus to Bitslice 6
  TX_BIT_CTRL_OUT0 => TX_BIT_CTRL_OUT0,    -- 40-bit output: Output bus to Bitslice 0
  TX_BIT_CTRL_OUT1 => TX_BIT_CTRL_OUT1,    -- 40-bit output: Output bus to Bitslice 1
  TX_BIT_CTRL_OUT2 => TX_BIT_CTRL_OUT2,    -- 40-bit output: Output bus to Bitslice 2
  TX_BIT_CTRL_OUT3 => TX_BIT_CTRL_OUT3,    -- 40-bit output: Output bus to Bitslice 3
  TX_BIT_CTRL_OUT4 => TX_BIT_CTRL_OUT4,    -- 40-bit output: Output bus to Bitslice 4
  TX_BIT_CTRL_OUT5 => TX_BIT_CTRL_OUT5,    -- 40-bit output: Output bus to Bitslice 5
  TX_BIT_CTRL_OUT6 => TX_BIT_CTRL_OUT6,    -- 40-bit output: Output bus to Bitslice 6
  TX_BIT_CTRL_OUT_TRI => TX_BIT_CTRL_OUT_TRI, -- 40-bit output: Output bus to 3-state TX BITSlice_TRI
  VTC_RDY => VTC_RDY,                   -- 1-bit output: PHY calibration is complete
  CLK_FROM_EXT => CLK_FROM_EXT,          -- 1-bit input: Inter-byte clock coming from north or south
-- BITSlice_CONTROL

  EN_VTC => EN_VTC,                      -- 1-bit input: Enables voltage and temperature compensation
-- when High

  NCLK_NIBBLE_IN => NCLK_NIBBLE_IN,       -- 1-bit input: Intra-byte DQS strobes from other/clock
-- control block

  PCLK_NIBBLE_IN => PCLK_NIBBLE_IN,       -- 1-bit input: Intra-byte DQS strobes/clock from other
-- control block

  PHY_RDCS0 => PHY_RDCS0,                 -- 4-bit input: Rank select
  PHY_RDCS1 => PHY_RDCS1,                 -- 4-bit input: Rank select
  PHY_RDEN => PHY_RDEN,                   -- 4-bit input: Read burst enable when using a memory
-- interface

  PHY_WRCS0 => PHY_WRCS0,                 -- 4-bit input: Rank select
  PHY_WRCS1 => PHY_WRCS1,                 -- 4-bit input: Rank select
  PLL_CLK => PLL_CLK,                     -- 1-bit input: PLL clock input
  REFCLK => REFCLK,                       -- 1-bit input: Frequency reference clock for delay control

```

```

RIU_ADDR => RIU_ADDR,
RIU_CLK => RIU_CLK,
RIU_NIBBLE_SEL => RIU_NIBBLE_SEL,
RIU_WR_DATA => RIU_WR_DATA,
RIU_WR_EN => RIU_WR_EN,
RST => RST,
RX_BIT_CTRL_IN0 => RX_BIT_CTRL_IN0,
RX_BIT_CTRL_IN1 => RX_BIT_CTRL_IN1,
RX_BIT_CTRL_IN2 => RX_BIT_CTRL_IN2,
RX_BIT_CTRL_IN3 => RX_BIT_CTRL_IN3,
RX_BIT_CTRL_IN4 => RX_BIT_CTRL_IN4,
RX_BIT_CTRL_IN5 => RX_BIT_CTRL_IN5,
RX_BIT_CTRL_IN6 => RX_BIT_CTRL_IN6,
TBYTE_IN => TBYTE_IN,
TX_BIT_CTRL_IN0 => TX_BIT_CTRL_IN0,
TX_BIT_CTRL_IN1 => TX_BIT_CTRL_IN1,
TX_BIT_CTRL_IN2 => TX_BIT_CTRL_IN2,
TX_BIT_CTRL_IN3 => TX_BIT_CTRL_IN3,
TX_BIT_CTRL_IN4 => TX_BIT_CTRL_IN4,
TX_BIT_CTRL_IN5 => TX_BIT_CTRL_IN5,
TX_BIT_CTRL_IN6 => TX_BIT_CTRL_IN6,
TX_BIT_CTRL_IN_TRI => TX_BIT_CTRL_IN_TRI
);

-- End of BITSlice_CONTROL_inst instantiation
    
```

## Verilog Instantiation Template

```

// BITSlice_CONTROL: BITSlice_CONTROL for control using Native Mode
// UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

BITSlice_CONTROL #(
    .DIV_MODE("DIV2"), // Controller DIV2/DIV4 mode (DIV2, DIV4)
    .EN_CLK_TO_EXT_NORTH("DISABLE"), // Enable clock forwarding to north
    .EN_CLK_TO_EXT_SOUTH("DISABLE"), // Enable clock forwarding to south
    .EN_DYN_ODLY_MODE("FALSE"), // Enable dynamic output delay mode
    .EN_OTHER_NCLK("FALSE"), // Select the NCLK from the other BITSlice_CONTROL in the nibble (FALSE,
    // TRUE)
    .EN_OTHER_PCLK("FALSE"), // Select the PCLK from the other BITSlice_CONTROL in the nibble (FALSE,
    // TRUE)
    .IDLY_VT_TRACK("TRUE"), // Enable VT tracking for input delays
    .INV_RXCLK("FALSE"), // Invert clock path from IOB to upper RX bitslice
    .ODLY_VT_TRACK("TRUE"), // Enable VT tracking for output delays
    .QDLY_VT_TRACK("TRUE"), // Enable VT tracking for clock delays
    .READ_IDLE_COUNT(6'h00), // Gap count between read bursts for ODT control counter (0-3f)
    .REFCLK_SRC("PLLCLK"), // Select the input clock for the delay control (PLLCLK, REFCLK)
    .ROUNDING_FACTOR(16), // Rounding factor in BISC spec (128-8)
    .RXGATE_EXTEND("FALSE"), // Reserved for use by MIG Memory Controller. Do Not Change.
    .RX_CLK_PHASE_N("SHIFT_0"), // Shift the Read CLK relative to read DQ during calibration (SHIFT_0,
    // SHIFT_90)
    .RX_CLK_PHASE_P("SHIFT_0"), // Shift the Read CLK relative to read DQ during calibration (SHIFT_0,
    // SHIFT_90)
    .RX_GATING("DISABLE"), // ENABLE/DISABLE read DQS gating
    .SELF_CALIBRATE("ENABLE"), // Enable BISC of nibble controlled by BITSlice_CONTROL
    .SERIAL_MODE("FALSE"), // Put BITSlice read paths into serial mode (FALSE, TRUE)
    .SIM_DEVICE("ULTRASCALE"), // Set the device version (ULTRASCALE, ULTRASCALE_PLUS_ES1)
    .TX_GATING("DISABLE") // ENABLE/DISABLE clock gating in WClkgen
)
BITSlice_CONTROL_inst (
    .CLK_TO_EXT_NORTH(CLK_TO_EXT_NORTH), // 1-bit output: Inter-byte clock going to north
    // BITSlice_CONTROL
    .CLK_TO_EXT_SOUTH(CLK_TO_EXT_SOUTH), // 1-bit output: Inter-byte clock going to south
    // BITSlice_CONTROL
    .DLY_RDY(DLY_RDY), // 1-bit output: Fixed delay calibration complete
    .DYN_DCI(DYN_DCI), // 7-bit output: Direct control of IOB DCI when using a memory
    // interface
    
```

```

.NCLK_NIBBLE_OUT(NCLK_NIBBLE_OUT), // 1-bit output: Intra-byte DQS strobes/clock to other control
// block

.PCLK_NIBBLE_OUT(PCLK_NIBBLE_OUT), // 1-bit output: Intra-byte DQS strobes/clock to other control
// block

.RIU_RD_DATA(RIU_RD_DATA), // 16-bit output: RIU Output Read data to the controller
.RIU_VALID(RIU_VALID), // 1-bit output: Last data written has been accepted when High
.RX_BIT_CTRL_OUT0(RX_BIT_CTRL_OUT0), // 40-bit output: Output bus to Bitslice 0
.RX_BIT_CTRL_OUT1(RX_BIT_CTRL_OUT1), // 40-bit output: Output bus to Bitslice 1
.RX_BIT_CTRL_OUT2(RX_BIT_CTRL_OUT2), // 40-bit output: Output bus to Bitslice 2
.RX_BIT_CTRL_OUT3(RX_BIT_CTRL_OUT3), // 40-bit output: Output bus to Bitslice 3
.RX_BIT_CTRL_OUT4(RX_BIT_CTRL_OUT4), // 40-bit output: Output bus to Bitslice 4
.RX_BIT_CTRL_OUT5(RX_BIT_CTRL_OUT5), // 40-bit output: Output bus to Bitslice 5
.RX_BIT_CTRL_OUT6(RX_BIT_CTRL_OUT6), // 40-bit output: Output bus to Bitslice 6
.TX_BIT_CTRL_OUT0(TX_BIT_CTRL_OUT0), // 40-bit output: Output bus to Bitslice 0
.TX_BIT_CTRL_OUT1(TX_BIT_CTRL_OUT1), // 40-bit output: Output bus to Bitslice 1
.TX_BIT_CTRL_OUT2(TX_BIT_CTRL_OUT2), // 40-bit output: Output bus to Bitslice 2
.TX_BIT_CTRL_OUT3(TX_BIT_CTRL_OUT3), // 40-bit output: Output bus to Bitslice 3
.TX_BIT_CTRL_OUT4(TX_BIT_CTRL_OUT4), // 40-bit output: Output bus to Bitslice 4
.TX_BIT_CTRL_OUT5(TX_BIT_CTRL_OUT5), // 40-bit output: Output bus to Bitslice 5
.TX_BIT_CTRL_OUT6(TX_BIT_CTRL_OUT6), // 40-bit output: Output bus to Bitslice 6
.TX_BIT_CTRL_OUT_TRI(TX_BIT_CTRL_OUT_TRI), // 40-bit output: Output bus to 3-state TX_BITSLICE_TRI
.VTC_RDY(VTC_RDY), // 1-bit output: PHY calibration is complete
.CLK_FROM_EXT(CLK_FROM_EXT), // 1-bit input: Inter-byte clock coming from north or south
// BITSLICE_CONTROL

.EN_VTC(EN_VTC), // 1-bit input: Enables voltage and temperature compensation
// when High

.NCLK_NIBBLE_IN(NCLK_NIBBLE_IN), // 1-bit input: Intra-byte DQS strobes from other/clock
// control block

.PCLK_NIBBLE_IN(PCLK_NIBBLE_IN), // 1-bit input: Intra-byte DQS strobes/clock from other
// control block

.PHY_RDCS0(PHY_RDCS0), // 4-bit input: Rank select
.PHY_RDCS1(PHY_RDCS1), // 4-bit input: Rank select
.PHY_RDEN(PHY_RDEN), // 4-bit input: Read burst enable when using a memory interface
.PHY_WRCS0(PHY_WRCS0), // 4-bit input: Rank select
.PHY_WRCS1(PHY_WRCS1), // 4-bit input: Rank select
.PLL_CLK(PLL_CLK), // 1-bit input: PLL clock input
.REFCLK(REFCLK), // 1-bit input: Frequency reference clock for delay control
.RIU_ADDR(RIU_ADDR), // 6-bit input: Address input for RIU
.RIU_CLK(RIU_CLK), // 1-bit input: System clock from fabric for RIU access
.RIU_NIBBLE_SEL(RIU_NIBBLE_SEL), // 1-bit input: Nibble select to enable RIU read/write
.RIU_WR_DATA(RIU_WR_DATA), // 16-bit input: RIU Input Write data from the controller
.RIU_WR_EN(RIU_WR_EN), // 1-bit input: Enables write to RIU when High
.RST(RST), // 1-bit input: Asynchronous global reset
.RX_BIT_CTRL_IN0(RX_BIT_CTRL_IN0), // 40-bit input: Input bus from Bitslice 0
.RX_BIT_CTRL_IN1(RX_BIT_CTRL_IN1), // 40-bit input: Input bus from Bitslice 1
.RX_BIT_CTRL_IN2(RX_BIT_CTRL_IN2), // 40-bit input: Input bus from Bitslice 2
.RX_BIT_CTRL_IN3(RX_BIT_CTRL_IN3), // 40-bit input: Input bus from Bitslice 3
.RX_BIT_CTRL_IN4(RX_BIT_CTRL_IN4), // 40-bit input: Input bus from Bitslice 4
.RX_BIT_CTRL_IN5(RX_BIT_CTRL_IN5), // 40-bit input: Input bus from Bitslice 5
.RX_BIT_CTRL_IN6(RX_BIT_CTRL_IN6), // 40-bit input: Input bus from Bitslice 6
.TBYTE_IN(TBYTE_IN), // 4-bit input: Output enable for 3-state control
.TX_BIT_CTRL_IN0(TX_BIT_CTRL_IN0), // 40-bit input: Input bus from Bitslice 0
.TX_BIT_CTRL_IN1(TX_BIT_CTRL_IN1), // 40-bit input: Input bus from Bitslice 1
.TX_BIT_CTRL_IN2(TX_BIT_CTRL_IN2), // 40-bit input: Input bus from Bitslice 2
.TX_BIT_CTRL_IN3(TX_BIT_CTRL_IN3), // 40-bit input: Input bus from Bitslice 3
.TX_BIT_CTRL_IN4(TX_BIT_CTRL_IN4), // 40-bit input: Input bus from Bitslice 4
.TX_BIT_CTRL_IN5(TX_BIT_CTRL_IN5), // 40-bit input: Input bus from Bitslice 5
.TX_BIT_CTRL_IN6(TX_BIT_CTRL_IN6), // 40-bit input: Input bus from Bitslice 6
.TX_BIT_CTRL_IN_TRI(TX_BIT_CTRL_IN_TRI) // 40-bit input: Input bus from 3-state TX_BITSLICE_TRI
);

// End of BITSLICE_CONTROL_inst instantiation

```

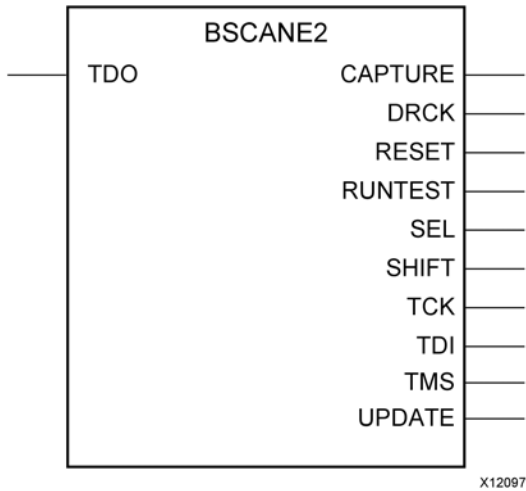
## For More Information

- See the *UltraScale Architecture SelectIO Resources User Guide* ([UG571](#)).
- See the [UltraScale Programmable Devices Documentation](#).

# BSCANE2

## Primitive: Boundary-Scan User Instruction

PRIMITIVE\_GROUP: CONFIGURATION  
 PRIMITIVE\_SUBGROUP: BSCAN  
 Families: UltraScale, UltraScale+



## 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 device. 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 instructions, instantiate four of these elements and set the JTAG\_CHAIN attribute appropriately.

For specific information on boundary scan for an architecture, see the Configuration User Guide for the specific device.

## Port Descriptions

Port	Direction	Width	Function
CAPTURE	Output	1	CAPTURE output from TAP controller.
DRCK	Output	1	Gated TCK output. When SEL is asserted, DRCK toggles when CAPTURE or SHIFT are asserted.
RESET	Output	1	Reset output for TAP controller.
RUNTEST	Output	1	Output asserted when TAP controller is in Run Test/Idle state.
SEL	Output	1	USER instruction active output.
SHIFT	Output	1	SHIFT output from TAP controller.
TCK	Output	1	Test Clock output. Fabric connection to TAP Clock pin.

Port	Direction	Width	Function
TDI	Output	1	Test Data Input (TDI) output from TAP controller.
TDO	Input	1	Test Data Output (TDO) input for USER function.
TMS	Output	1	Test Mode Select output. Fabric connection to TAP.
UPDATE	Output	1	UPDATE output from TAP controller

## Design Entry Method

Instantiation	Recommended
Inference	No
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
JTAG_CHAIN	DECIMAL	1, 2, 3, 4	1	Value for USER command

## VHDL Instantiation Template

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- BSCANE2: Boundary-Scan User Instruction
--      UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

BSCANE2_inst : BSCANE2
generic map (
    JTAG_CHAIN => 1 -- Value for USER command
)
port map (
    CAPTURE => CAPTURE, -- 1-bit output: CAPTURE output from TAP controller.
    DRCK => DRCK,      -- 1-bit output: Gated TCK output. When SEL is asserted, DRCK toggles when CAPTURE or
                        -- SHIFT are asserted.

    RESET => RESET,    -- 1-bit output: Reset output for TAP controller.
    RUNTEST => RUNTEST, -- 1-bit output: Output asserted when TAP controller is in Run Test/Idle state.
    SEL => SEL,        -- 1-bit output: USER instruction active output.
    SHIFT => SHIFT,    -- 1-bit output: SHIFT output from TAP controller.
    TCK => TCK,        -- 1-bit output: Test Clock output. Fabric connection to TAP Clock pin.
    TDI => TDI,        -- 1-bit output: Test Data Input (TDI) output from TAP controller.
    TMS => TMS,        -- 1-bit output: Test Mode Select output. Fabric connection to TAP.
    UPDATE => UPDATE,  -- 1-bit output: UPDATE output from TAP controller
    TDO => TDO         -- 1-bit input: Test Data Output (TDO) input for USER function.
);

-- End of BSCANE2_inst instantiation
    
```



## Verilog Instantiation Template

```

// BSCANE2: Boundary-Scan User Instruction
// UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

BSCANE2 #(
    .JTAG_CHAIN(1) // Value for USER command
)
BSCANE2_inst (
    .CAPTURE(CAPTURE), // 1-bit output: CAPTURE output from TAP controller.
    .DRCK(DRCK),       // 1-bit output: Gated TCK output. When SEL is asserted, DRCK toggles when CAPTURE or
                        // SHIFT are asserted.

    .RESET(RESET),     // 1-bit output: Reset output for TAP controller.
    .RUNTEST(RUNTEST), // 1-bit output: Output asserted when TAP controller is in Run Test/Idle state.
    .SEL(SEL),         // 1-bit output: USER instruction active output.
    .SHIFT(SHIFT),     // 1-bit output: SHIFT output from TAP controller.
    .TCK(TCK),        // 1-bit output: Test Clock output. Fabric connection to TAP Clock pin.
    .TDI(TDI),        // 1-bit output: Test Data Input (TDI) output from TAP controller.
    .TMS(TMS),        // 1-bit output: Test Mode Select output. Fabric connection to TAP.
    .UPDATE(UPDATE),  // 1-bit output: UPDATE output from TAP controller
    .TDO(TDO)         // 1-bit input: Test Data Output (TDO) input for USER function.
);

// End of BSCANE2_inst instantiation

```

## For More Information

- See the *UltraScale Architecture Configuration User Guide* ([UG570](#)).
- See the [UltraScale Programmable Devices Documentation](#).

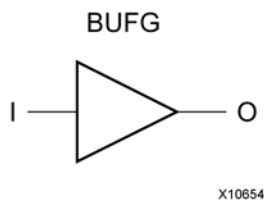
## BUFG

Primitive: General Clock Buffer

PRIMITIVE\_GROUP: CLOCK

PRIMITIVE\_SUBGROUP: BUFFER

Families: UltraScale, UltraScale+



### 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 input
O	Output	1	Clock output

### Design Entry Method

Instantiation	Yes
Inference	Recommended
IP and IP Integrator Catalog	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: General Clock Buffer
-- UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

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

```
-- End of BUFG_inst instantiation
```

## Verilog Instantiation Template

```
// BUFG: General Clock Buffer
//      UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

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

// End of BUFG_inst instantiation
```

## For More Information

- See the *UltraScale Architecture Clocking Resources User Guide* ([UG572](#)).
- See the [UltraScale Programmable Devices Documentation](#).

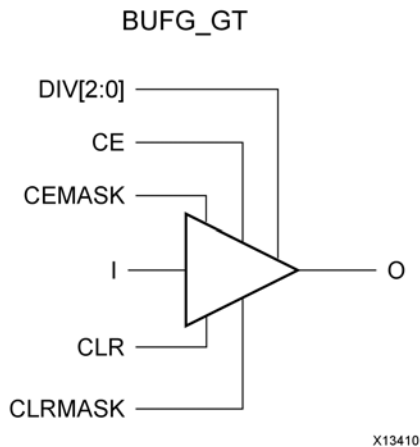
## BUFG\_GT

Primitive: Clock Buffer Driven by Gigabit Transceiver

PRIMITIVE\_GROUP: CLOCK

PRIMITIVE\_SUBGROUP: BUFFER

Families: UltraScale, UltraScale+



### Introduction

Clock buffer driven by the Gigabit transceiver for the purpose of clock distribution to other portions of the device.

### Port Descriptions

Port	Direction	Width	Function
CE	Input	1	Buffer enable
CEMASK	Input	1	CE Mask
CLR	Input	1	Asynchronous clear forcing the output to zero.
CLRMASK	Input	1	CLR Mask.
DIV<2:0>	Input	3	Specifies the value to divide the clock. Divide value is value provided plus 1. For instance, setting 3'b000 will provide a divide value of 1 and 3'b111 will be a divide value of 8.
I	Input	1	Buffer input.
O	Output	1	Buffer output

### Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	Recommended

## VHDL 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_GT: Clock Buffer Driven by Gigabit Transceiver
--           UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

BUFG_GT_inst : BUFG_GT
port map (
    O => O,           -- 1-bit output: Buffer
    CE => CE,         -- 1-bit input: Buffer enable
    CEMASK => CEMASK, -- 1-bit input: CE Mask
    CLR => CLR,       -- 1-bit input: Asynchronous clear
    CLRMASK => CLRMASK, -- 1-bit input: CLR Mask
    DIV => DIV,      -- 3-bit input: Dynamic divide Value
    I => I           -- 1-bit input: Buffer
);

-- End of BUFG_GT_inst instantiation
    
```

## Verilog Instantiation Template

```

// BUFG_GT: Clock Buffer Driven by Gigabit Transceiver
//           UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

BUFG_GT BUFG_GT_inst (
    .O(O),           // 1-bit output: Buffer
    .CE(CE),        // 1-bit input: Buffer enable
    .CEMASK(CEMASK), // 1-bit input: CE Mask
    .CLR(CLR),      // 1-bit input: Asynchronous clear
    .CLRMASK(CLRMASK), // 1-bit input: CLR Mask
    .DIV(DIV),      // 3-bit input: Dynamic divide Value
    .I(I)           // 1-bit input: Buffer
);

// End of BUFG_GT_inst instantiation
    
```

## For More Information

- See the *UltraScale Architecture Clocking Resources User Guide* ([UG572](#)).
- See the [UltraScale Programmable Devices Documentation](#).

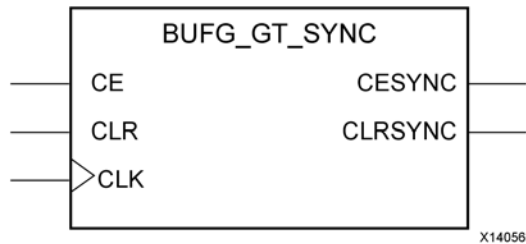
## BUFG\_GT\_SYNC

Primitive: Synchronizer for BUFG\_GT Control Signals

PRIMITIVE\_GROUP: CLOCK

PRIMITIVE\_SUBGROUP: CLOCK\_SYNC

Families: UltraScale, UltraScale+



### Introduction

Synchronizer for the BUFG\_GT CE and CLR functions. See the UltraScale GT Transceivers User Guide for details on this component.

### Port Descriptions

Port	Direction	Width	Function
CE	Input	1	Asynchronous enable
CESYNC	Output	1	CE signal synchronized to CLK
CLK	Input	1	Clock
CLR	Input	1	Asynchronous clear
CLRSYNC	Output	1	CLR signal synchronized to CLK

### Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	Recommended

## VHDL 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_GT_SYNC: Synchronizer for BUFG_GT Control Signals
-- UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

BUFG_GT_SYNC_inst : BUFG_GT_SYNC
port map (
  CESYNC => CESYNC,    -- 1-bit output: Synchronized CE
  CLRSYNC => CLRSYNC,  -- 1-bit output: Synchronized CLR
  CE => CE,            -- 1-bit input: Asynchronous enable
  CLK => CLK,          -- 1-bit input: Clock
  CLR => CLR           -- 1-bit input: Asynchronous clear
);

-- End of BUFG_GT_SYNC_inst instantiation
  
```

## Verilog Instantiation Template

```

// BUFG_GT_SYNC: Synchronizer for BUFG_GT Control Signals
// UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

BUFG_GT_SYNC BUFG_GT_SYNC_inst (
  .CESYNC(CESYNC),    // 1-bit output: Synchronized CE
  .CLRSYNC(CLRSYNC), // 1-bit output: Synchronized CLR
  .CE(CE),           // 1-bit input: Asynchronous enable
  .CLK(CLK),         // 1-bit input: Clock
  .CLR(CLR)          // 1-bit input: Asynchronous clear
);

// End of BUFG_GT_SYNC_inst instantiation
  
```

## For More Information

- See the *UltraScale Architecture Clocking Resources User Guide* ([UG572](#)).
- See the [UltraScale Programmable Devices Documentation](#).

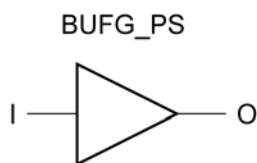
## BUFG\_PS

**Primitive:** A high-fanout buffer for low-skew distribution of the PS Clock signals

PRIMITIVE\_GROUP: [CLOCK](#)

PRIMITIVE\_SUBGROUP: [BUFFER](#)

Families: UltraScale+



X15112-101115

### Introduction

A high-fanout buffer for low-skew distribution of the PS Clock signals

### Port Descriptions

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

### For More Information

- See the *Zynq UltraScale+ MPSoC Technical Reference Manual* ([UG1085](#)).
- See the [UltraScale Programmable Devices Documentation](#).



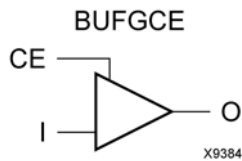
# BUFGCE

Primitive: General Clock Buffer with Clock Enable

PRIMITIVE\_GROUP: CLOCK

PRIMITIVE\_SUBGROUP: BUFFER

Families: UltraScale, UltraScale+



## Introduction

This design element is a general 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
CE	Input	1	Clock buffer active-High enable
I	Input	1	Buffer input
O	Output	1	Buffer output

## Design Entry Method

Instantiation	Recommended
Inference	Yes
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
CE_TYPE	STRING	"SYNC", "ASYN" "	"SYNC"	Specifies whether the enable should be synchronous (glitch-free) or asynchronous (no input clock switching necessary).
IS_CE_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies use of the programmable inversion on the CE pin.
IS_I_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies use of the programmable inversion on the I pin.

## VHDL 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: General Clock Buffer with Clock Enable
-- UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

BUFGCE_inst : BUFGCE
generic map (
    CE_TYPE => "SYNC",      -- ASYNC, SYNC
    IS_CE_INVERTED => '0', -- Programmable inversion on CE
    IS_I_INVERTED => '0'   -- Programmable inversion on I
)
port map (
    O => O,    -- 1-bit output: Buffer
    CE => CE,  -- 1-bit input: Buffer enable
    I => I     -- 1-bit input: Buffer
);

-- End of BUFGCE_inst instantiation
    
```

## Verilog Instantiation Template

```

// BUFGCE: General Clock Buffer with Clock Enable
// UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

BUFGCE #(
    .CE_TYPE("SYNC"),      // ASYNC, SYNC
    .IS_CE_INVERTED(1'b0), // Programmable inversion on CE
    .IS_I_INVERTED(1'b0)  // Programmable inversion on I
)
BUFGCE_inst (
    .O(O),    // 1-bit output: Buffer
    .CE(CE),  // 1-bit input: Buffer enable
    .I(I)     // 1-bit input: Buffer
);

// End of BUFGCE_inst instantiation
    
```

## For More Information

- See the *UltraScale Architecture Clocking Resources User Guide* ([UG572](#)).
- See the [UltraScale Programmable Devices Documentation](#).

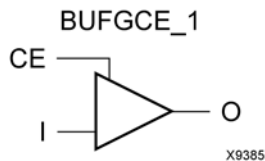
# BUFGCE\_1

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

PRIMITIVE\_GROUP: CLOCK

PRIMITIVE\_SUBGROUP: BUFFER

Families: UltraScale, UltraScale+



## Introduction

This design element is a general clock buffer with a single gated input. Its O output is 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
CE	Input	1	Clock buffer active-High enable
I	Input	1	Clock input
O	Output	1	Clock output

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	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: General Clock Buffer with Clock Enable and Output State 1
-- UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

BUFGCE_1_inst : BUFGCE_1
port map (
    O => O, -- 1-bit output: Clock output
    CE => CE, -- 1-bit input: Clock buffer active-High enable
    I => I -- 1-bit input: Clock input
);

-- End of BUFGCE_1_inst instantiation
```

## Verilog Instantiation Template

```
// BUFGCE_1: General Clock Buffer with Clock Enable and Output State 1
// UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

BUFGCE_1 BUFGCE_1_inst (
    .O(O), // 1-bit output: Clock output
    .CE(CE), // 1-bit input: Clock buffer active-High enable
    .I(I) // 1-bit input: Clock input
);

// End of BUFGCE_1_inst instantiation
```

## For More Information

- See the *UltraScale Architecture Clocking Resources User Guide* ([UG572](#)).
- See the [UltraScale Programmable Devices Documentation](#).

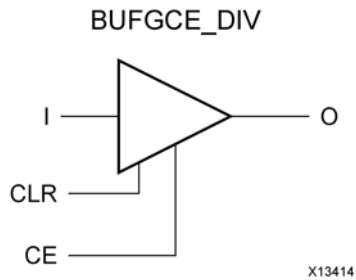
## BUFGCE\_DIV

Primitive: General Clock Buffer with Divide Function

PRIMITIVE\_GROUP: CLOCK

PRIMITIVE\_SUBGROUP: BUFFER

Families: UltraScale, UltraScale+



### Introduction

BUFGCE\_DIV is a general clock buffer with an enable and divide function.

### Port Descriptions

Port	Direction	Width	Function
CE	Input	1	Buffer enable input.
CLR	Input	1	Asynchronous clear function forcing the output value to zero.
I	Input	1	Buffer input
O	Output	1	Buffer output

### Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	No

### Available Attributes

Attribute	Type	Allowed Values	Default	Description
BUFGCE_DIVIDE	DECIMAL	1, 2, 3, 4, 5, 6, 7, 8	1	Divide value.
<p><b>Programmable Inversion Attributes:</b> Specifies whether or not to use the optional inversions are to be used on specific pins for this component to change the active polarity of the pin function. When set to 1, it changes the function to behave active-Low rather than active-High. If an external inverter is specified on one of these associated pins, the Vivado Design Suite will automatically set this attribute during the opt_design stage so that additional logic is not necessary for changing the input polarity.</p>				

Attribute	Type	Allowed Values	Default	Description
IS_CE_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the CE pin of this component.
IS_CLR_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the CLR pin of this component.
IS_I_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the I pin of this component.

## VHDL Instantiation Template

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- BUFGCE_DIV: General Clock Buffer with Divide Function
--           UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

BUFGCE_DIV_inst : BUFGCE_DIV
generic map (
    BUFGCE_DIVIDE => 1,      -- 1-8
    -- Programmable Inversion Attributes: Specifies built-in programmable inversion on specific pins
    IS_CE_INVERTED => '0',  -- Optional inversion for CE
    IS_CLR_INVERTED => '0', -- Optional inversion for CLR
    IS_I_INVERTED  => '0'   -- Optional inversion for I
)
port map (
    O => O,      -- 1-bit output: Buffer
    CE => CE,    -- 1-bit input: Buffer enable
    CLR => CLR,  -- 1-bit input: Asynchronous clear
    I => I       -- 1-bit input: Buffer
);

-- End of BUFGCE_DIV_inst instantiation
    
```

## Verilog Instantiation Template

```

// BUFGCE_DIV: General Clock Buffer with Divide Function
//           UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

BUFGCE_DIV #(
    .BUFGCE_DIVIDE(1),      // 1-8
    // Programmable Inversion Attributes: Specifies built-in programmable inversion on specific pins
    .IS_CE_INVERTED(1'b0), // Optional inversion for CE
    .IS_CLR_INVERTED(1'b0), // Optional inversion for CLR
    .IS_I_INVERTED(1'b0)   // Optional inversion for I
)
BUFGCE_DIV_inst (
    .O(O),      // 1-bit output: Buffer
    .CE(CE),   // 1-bit input: Buffer enable
    .CLR(CLR), // 1-bit input: Asynchronous clear
    .I(I)      // 1-bit input: Buffer
);

// End of BUFGCE_DIV_inst instantiation
    
```

## For More Information

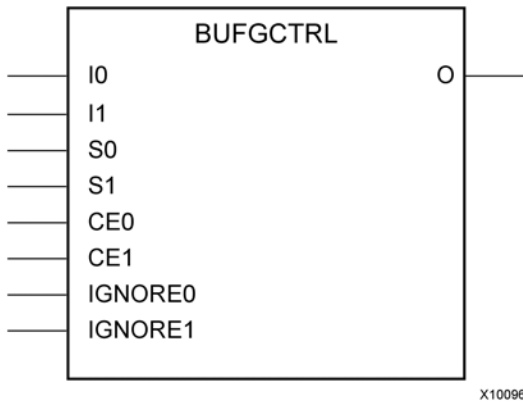
- See the *UltraScale Architecture Clocking Resources User Guide* ([UG572](#)).
- See the [UltraScale Programmable Devices Documentation](#).



# BUFGCTRL

Primitive: General Clock Control Buffer

PRIMITIVE\_GROUP: CLOCK  
 PRIMITIVE\_SUBGROUP: MUX  
 Families: UltraScale, UltraScale+



## Introduction

BUFGCTRL primitive is a general clock buffer that is designed as a synchronous/asynchronous "glitch free" 2:1 multiplexer with two clock inputs. If clock multiplexing is not necessary, you should use a BUFG or BUFGCE component.

## Port Descriptions

Port	Direction	Width	Function
CE0	Input	1	Clock enable input for the I0 clock input. A setup/hold time must be guaranteed when you are using the CE0 pin to enable this input. Failure to meet this requirement could result in a clock glitch.
CE1	Input	1	Clock enable input for the I1 clock input. A setup/hold time must be guaranteed when you are using the CE1 pin to enable this input. Failure to meet this requirement could result in a clock glitch.
IGNORE0	Input	1	Clock ignore input for I0 input. Asserting the IGNORE pin prevents the BUFGCTRL from detecting the conditions for switching between two clock inputs. In other words, asserting IGNORE causes the MUX to switch the inputs at the instant the select pin changes. IGNORE0 causes the output to switch away from the I0 input immediately when the select pin changes, while IGNORE1 causes the output to switch away from the I1 input immediately when the select pin changes.
IGNORE1	Input	1	Clock ignore input for I1 input. Asserting the IGNORE pin prevents the BUFGCTRL from detecting the conditions for switching between two clock inputs. In other words, asserting IGNORE causes the MUX to switch the inputs at the instant the select pin changes. IGNORE0 causes the output to switch away from the I0 input immediately when the select pin changes, while IGNORE1 causes the output to switch away from the I1 input immediately when the select pin changes.

Port	Direction	Width	Function
I0	Input	1	Primary clock input into the BUFGCTRL enabled by the CE0 input and selected by the S0 input.
I1	Input	1	Secondary clock input into the BUFGCTRL enabled by the CE1 input and selected by the S1 input.
O	Output	1	Clock output
S0	Input	1	Clock select input for I0. The S pins represent the clock select pin for each clock input. 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.
S1	Input	1	Clock select input for I1. The S pins represent the clock select pin for each clock input. 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.

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT_OUT	DECIMAL	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.
<b>Programmable Inversion Attributes:</b> Specifies whether or not to use the optional inversion on specific pins for this component to change the active polarity of the pin function. When set to 1, it changes the function to behave active-Low rather than active-High. If an external inverter is specified on one of these associated pins, the Vivado Design Suite will automatically set this attribute during the opt_design stage so that additional logic is not necessary for changing the input polarity.				
IS_CE0_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the CE0 pin of this component.
IS_CE1_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the CE1 pin of this component.
IS_IGNORE0_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the IGNORE0 pin of this component.
IS_IGNORE1_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the IGNORE1 pin of this component.

Attribute	Type	Allowed Values	Default	Description
IS_I0_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the I0 pin of this component.
IS_I1_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the I1 pin of this component.
IS_S0_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the S0 pin of this component.
IS_S1_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the S1 pin of this component.

**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: General Clock Control Buffer
--           UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

BUFGCTRL_inst : BUFGCTRL
generic map (
    INIT_OUT => 0,                -- Initial value of BUFGCTRL output, 0-1
    PRESELECT_I0 => FALSE,        -- BUFGCTRL output uses I0 input, FALSE, TRUE
    PRESELECT_I1 => FALSE,        -- BUFGCTRL output uses I1 input, FALSE, TRUE
    -- Programmable Inversion Attributes: Specifies built-in programmable inversion on specific pins
    IS_CE0_INVERTED => '0',      -- Optional inversion for CE0
    IS_CE1_INVERTED => '0',      -- Optional inversion for CE1
    IS_I0_INVERTED => '0',       -- Optional inversion for I0
    IS_I1_INVERTED => '0',       -- Optional inversion for I1
    IS_IGNORE0_INVERTED => '0',  -- Optional inversion for IGNORE0
    IS_IGNORE1_INVERTED => '0',  -- Optional inversion for IGNORE1
    IS_S0_INVERTED => '0',       -- Optional inversion for S0
    IS_S1_INVERTED => '0',       -- Optional inversion for S1
)
port map (
    O => O,                -- 1-bit output: Clock output
    CE0 => CE0,            -- 1-bit input: Clock enable input for I0
    CE1 => CE1,            -- 1-bit input: Clock enable input for I1
    I0 => I0,              -- 1-bit input: Primary clock
    I1 => I1,              -- 1-bit input: Secondary clock
    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 for I0
    S1 => S1,              -- 1-bit input: Clock select for I1
);

-- End of BUFGCTRL_inst instantiation
    
```

## Verilog Instantiation Template

```
// BUFGCTRL: General Clock Control Buffer
// UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

BUFGCTRL #(
    .INIT_OUT(0),                // Initial value of BUFGCTRL output, 0-1
    .PRESELECT_I0("FALSE"),     // BUFGCTRL output uses I0 input, FALSE, TRUE
    .PRESELECT_I1("FALSE"),     // BUFGCTRL output uses I1 input, FALSE, TRUE
    // Programmable Inversion Attributes: Specifies built-in programmable inversion on specific pins
    .IS_CE0_INVERTED(1'b0),     // Optional inversion for CE0
    .IS_CE1_INVERTED(1'b0),     // Optional inversion for CE1
    .IS_I0_INVERTED(1'b0),      // Optional inversion for I0
    .IS_I1_INVERTED(1'b0),      // Optional inversion for I1
    .IS_IGNORE0_INVERTED(1'b0), // Optional inversion for IGNORE0
    .IS_IGNORE1_INVERTED(1'b0), // Optional inversion for IGNORE1
    .IS_S0_INVERTED(1'b0),      // Optional inversion for S0
    .IS_S1_INVERTED(1'b0)       // Optional inversion for S1
)
BUFGCTRL_inst (
    .O(O),                       // 1-bit output: Clock output
    .CE0(CE0),                   // 1-bit input: Clock enable input for I0
    .CE1(CE1),                   // 1-bit input: Clock enable input for I1
    .I0(I0),                     // 1-bit input: Primary clock
    .I1(I1),                     // 1-bit input: Secondary clock
    .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 for I0
    .S1(S1)                      // 1-bit input: Clock select for I1
);

// End of BUFGCTRL_inst instantiation
```

## For More Information

- See the *UltraScale Architecture Clocking Resources User Guide* ([UG572](#)).
- See the [UltraScale Programmable Devices Documentation](#).

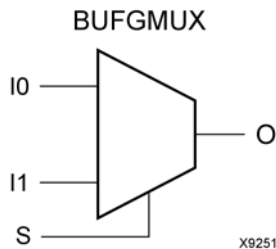
# BUFGMUX

Primitive: General Clock Mux Buffer

PRIMITIVE\_GROUP: CLOCK

PRIMITIVE\_SUBGROUP: MUX

Families: UltraScale, UltraScale+



## Introduction

This design element is a general clock buffer, based off of the BUFGCTRL, that can select between two input clocks: I0 and I1. When the select input (S) is Low, the signal on I0 is selected for output (O). When the select input (S) is High, the signal on I1 is selected for output. BUFGMUX and BUFGMUX\_1 are distinguished by the state the output assumes when that output switches between clocks in response to a change in its select input. BUFGMUX assumes output state 0 and BUFGMUX\_1 assumes output state 1.

## Port Descriptions

Port	Direction	Width	Function
I0	Input	1	Clock buffer input. This input is reflected on the output O when the S input is zero.
I1	Input	1	Clock buffer input. This input is reflected on the output O when the S input is one.
O	Output	1	Clock buffer output.
S	Input	1	Clock buffer select input. When Low, selects I0 input and when High, the I1 input is selected

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
CLK_SEL_TYPE	STRING	"SYNC", "ASYNC"	"SYNC"	Specifies synchronous (glitch-free) or asynchronous clock switching.

## VHDL 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: General Clock Mux Buffer
--           UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

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

-- End of BUFGMUX_inst instantiation
    
```

## Verilog Instantiation Template

```

// BUFGMUX: General Clock Mux Buffer
//           UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

BUFGMUX #(
    .CLK_SEL_TYPE("SYNC") // ASYNC, SYNC
)
BUFGMUX_inst (
    .O(O), // 1-bit output: Clock output
    .I0(I0), // 1-bit input: Clock input (S=0)
    .I1(I1), // 1-bit input: Clock input (S=1)
    .S(S) // 1-bit input: Clock select
);

// End of BUFGMUX_inst instantiation
    
```

## For More Information

- See the *UltraScale Architecture Clocking Resources User Guide* ([UG572](#)).
- See the [UltraScale Programmable Devices Documentation](#).

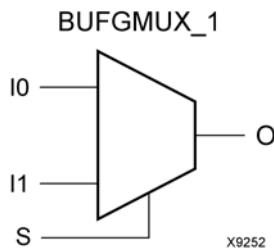
# BUFGMUX\_1

Primitive: General Clock Mux Buffer with Output State 1

PRIMITIVE\_GROUP: CLOCK

PRIMITIVE\_SUBGROUP: MUX

Families: UltraScale, UltraScale+



## Introduction

This design element is a general clock buffer, based off of the BUFGCTRL, that can select between two input clocks: I0 and I1. When the select input (S) is Low, the signal on I0 is selected for output (O). When the select input (S) is High, the signal on I1 is selected for output. BUFGMUX and BUFGMUX\_1 are distinguished by the state the output assumes when that output switches between clocks in response to a change in its select input. BUFGMUX assumes output state0 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

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
CLK_SEL_TYPE	STRING	"SYNC", "ASYN"	"SYNC"	Specifies synchronous (glitch-free) or asynchronous clock switching.

## VHDL 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: General Clock Mux Buffer with Output State 1
--           UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

BUFGMUX_1_inst : BUFGMUX_1
generic map (
    CLK_SEL_TYPE => "SYNC" -- ASYNC, SYNC
)
port map (
    O => O,    -- 1-bit output: Clock output
    I0 => I0,  -- 1-bit input: Clock input (S=0)
    I1 => I1,  -- 1-bit input: Clock input (S=1)
    S => S     -- 1-bit input: Clock select
);

-- End of BUFGMUX_1_inst instantiation
    
```

## Verilog Instantiation Template

```

// BUFGMUX_1: General Clock Mux Buffer with Output State 1
//           UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

BUFGMUX_1 #(
    .CLK_SEL_TYPE("SYNC") // ASYNC, SYNC
)
BUFGMUX_1_inst (
    .O(O), // 1-bit output: Clock output
    .I0(I0), // 1-bit input: Clock input (S=0)
    .I1(I1), // 1-bit input: Clock input (S=1)
    .S(S) // 1-bit input: Clock select
);

// End of BUFGMUX_1_inst instantiation
    
```

## For More Information

- See the *UltraScale Architecture Clocking Resources User Guide* ([UG572](#)).
- See the [UltraScale Programmable Devices Documentation](#).



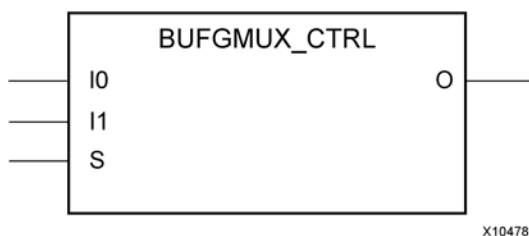
## BUFGMUX\_CTRL

Primitive: 2-to-1 General Clock MUX Buffer

PRIMITIVE\_GROUP: CLOCK

PRIMITIVE\_SUBGROUP: MUX

Families: UltraScale, UltraScale+



### Introduction

This design element is a general clock buffer with two clock inputs, one clock output, and a select line used to cleanly select between one of two clocks driving the clocking resources. 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
I0	Input	1	Clock buffer input. This input is reflected on the output O when the S input is zero.
I1	Input	1	Clock buffer input. This input is reflected on the output O when the S input is one.
O	Output	1	Clock buffer output.
S	Input	1	Clock buffer select input. When Low, selects the I0 input and when High, selects the I1 input.

### Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	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 General Clock MUX Buffer
-- UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

BUFGMUX_CTRL_inst : BUFGMUX_CTRL
port map (
    O => O,    -- 1-bit output: Clock output
    I0 => I0,  -- 1-bit input: Clock input (S=0)
    I1 => I1,  -- 1-bit input: Clock input (S=1)
    S => S     -- 1-bit input: Clock select
);

-- End of BUFGMUX_CTRL_inst instantiation
```

## Verilog Instantiation Template

```
// BUFGMUX_CTRL: 2-to-1 General Clock MUX Buffer
// UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

BUFGMUX_CTRL BUFGMUX_CTRL_inst (
    .O(O),    // 1-bit output: Clock output
    .I0(I0), // 1-bit input: Clock input (S=0)
    .I1(I1), // 1-bit input: Clock input (S=1)
    .S(S)     // 1-bit input: Clock select
);

// End of BUFGMUX_CTRL_inst instantiation
```

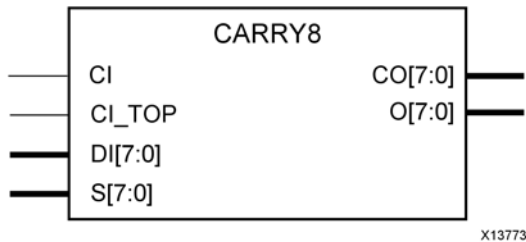
## For More Information

- See the *UltraScale Architecture Clocking Resources User Guide* ([UG572](#)).
- See the [UltraScale Programmable Devices Documentation](#).

# CARRY8

## Primitive: Fast Carry Logic with Look Ahead

PRIMITIVE\_GROUP: CLB  
 PRIMITIVE\_SUBGROUP: CARRY  
 Families: UltraScale, UltraScale+



## Introduction

This circuit represents the fast carry logic for a CLB. The carry chain consists of a series of eight MUXes and eight XORs that connect to the other logic (LUTs) in the CLB 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. This component can be configured to operate as a single 8-bit carry or two independent 4-bit carry logic.

## Port Descriptions

Port	Direction	Width	Function
CI	Input	1	Carry input for 8-bit carry or lower portion of 4-bit carry.
CI_TOP	Input	1	Upper carry input when CARRY_TYPE=DUAL_CY4. Tie to ground if CARRY_TYPE=SINGLE_CY8.
CO<7:0>	Output	8	Carry-out of each stage of the carry chain
DI<7:0>	Input	8	Carry-MUX data input
O<7:0>	Output	8	Carry chain XOR general data out
S<7:0>	Input	8	Carry-MUX select line

## Design Entry Method

Instantiation	Yes
Inference	Recommended
IP and IP Integrator Catalog	Yes

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
CARRY_TYPE	STRING	"SINGLE_CY8", "DUAL_CY4"	"SINGLE_CY8"	Specifies whether the CLB carry logic is to function as a single 8-bit carry-chain or two independent 4-bit carry chains.

## VHDL Instantiation Template

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- CARRY8: Fast Carry Logic with Look Ahead
--      UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

CARRY8_inst : CARRY8
generic map (
    CARRY_TYPE => "SINGLE_CY8" -- 8-bit or dual 4-bit carry (DUAL_CY4, SINGLE_CY8)
)
port map (
    CO => CO,           -- 8-bit output: Carry-out
    O => O,             -- 8-bit output: Carry chain XOR data out
    CI => CI,           -- 1-bit input: Lower Carry-In
    CI_TOP => CI_TOP,  -- 1-bit input: Upper Carry-In
    DI => DI,           -- 8-bit input: Carry-MUX data in
    S => S              -- 8-bit input: Carry-mux select
);

-- End of CARRY8_inst instantiation
    
```

## Verilog Instantiation Template

```

// CARRY8: Fast Carry Logic with Look Ahead
//      UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

CARRY8 #(
    .CARRY_TYPE("SINGLE_CY8") // 8-bit or dual 4-bit carry (DUAL_CY4, SINGLE_CY8)
)
CARRY8_inst (
    .CO(CO),           // 8-bit output: Carry-out
    .O(O),             // 8-bit output: Carry chain XOR data out
    .CI(CI),           // 1-bit input: Lower Carry-In
    .CI_TOP(CI_TOP),  // 1-bit input: Upper Carry-In
    .DI(DI),           // 8-bit input: Carry-MUX data in
    .S(S)              // 8-bit input: Carry-mux select
);

// End of CARRY8_inst instantiation
    
```

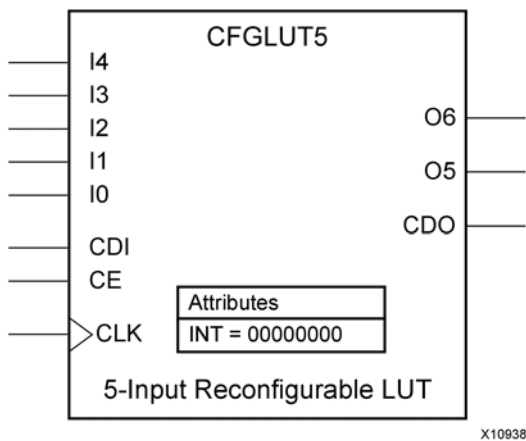
## For More Information

- See the *UltraScale Architecture Configurable Logic Block User Guide* ([UG574](#)).
- See the [UltraScale Programmable Devices Documentation](#).

# CFGLUT5

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

PRIMITIVE\_GROUP: CLB  
 PRIMITIVE\_SUBGROUP: LUT  
 Families: UltraScale, UltraScale+



## 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 eight LUT6 components within a CLB.

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
CDI	Input	1	Reconfiguration data serial input
CDO	Output	1	Reconfiguration data cascaded output (optionally connect to the CDI input of a subsequent LUT)
CE	Input	1	Active-High reconfiguration clock enable
CLK	Input	1	Reconfiguration clock
O5	Output	1	4-LUT output
O6	Output	1	5-LUT output
<b>LUT Inputs:</b> Logic inputs to the programmable look-up table			

Port	Direction	Width	Function
I0	Input	1	Logic inputs to the programmable look-up table
I1	Input	1	Logic inputs to the programmable look-up table
I2	Input	1	Logic inputs to the programmable look-up table
I3	Input	1	Logic inputs to the programmable look-up table
I4	Input	1	Logic inputs to the programmable look-up table

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	HEX	Any 32-bit HEX value	All zeroes	Specifies the initial logical expression of this element
IS_CLK_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Indicates whether the CLK is active-high or active-low

## VHDL 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: 5-input Dynamically Reconfigurable Look-Up Table (LUT)
-- UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

CFGLUT5_inst : CFGLUT5
generic map (
    INIT => X"00000000",    -- Initial logic function
    IS_CLK_INVERTED => '0' -- Optional inversion for CLK
)
port map (
    CDO => CDO, -- 1-bit output: Reconfiguration cascade
    O5 => O5,   -- 1-bit output: 4-LUT
    O6 => O6,   -- 1-bit output: 5-LUT
    CDI => CDI, -- 1-bit input: Reconfiguration data
    CE  => CE,  -- 1-bit input: Reconfiguration enable
    CLK => CLK, -- 1-bit input: Clock
    -- LUT Inputs inputs: Logic inputs
    I0 => I0,
    I1 => I1,
    I2 => I2,
    I3 => I3,
    I4 => I4
);

-- End of CFGLUT5_inst instantiation
    
```

## Verilog Instantiation Template

```
// CFGLUT5: 5-input Dynamically Reconfigurable Look-Up Table (LUT)
//           UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

CFGLUT5 #(
    .INIT(32'h00000000), // Initial logic function
    .IS_CLK_INVERTED(1'b0) // Optional inversion for CLK
)
CFGLUT5_inst (
    .CDO(CDO), // 1-bit output: Reconfiguration cascade
    .O5(O5), // 1-bit output: 4-LUT
    .O6(O6), // 1-bit output: 5-LUT
    .CDI(CDI), // 1-bit input: Reconfiguration data
    .CE(CE), // 1-bit input: Reconfiguration enable
    .CLK(CLK), // 1-bit input: Clock
    // LUT Inputs inputs: Logic inputs
    .I0(I0),
    .I1(I1),
    .I2(I2),
    .I3(I3),
    .I4(I4)
);

// End of CFGLUT5_inst instantiation
```

## For More Information

- See the *UltraScale Architecture Migration: Methodology Guide* ([UG1026](#)).
- See the [UltraScale Programmable Devices Documentation](#).

# CMAC

Primitive: 100G MAC Block

PRIMITIVE\_GROUP: **ADVANCED**

PRIMITIVE\_SUBGROUP: **MAC**

Families: UltraScale

## Introduction

The 100G MAC Block provides a high-performance, low latency 100G Ethernet port that allows for a wide range of user customization and statistics gathering. It supports 1588 time stamping for one step and two step. This element is not intended to be instantiated, used, or modified outside of Xilinx generated IP. The block is designed to be integrated with GTs and device clocking resources using fabric interconnect. See the UltraScale Architecture Integrated Block for 100G Ethernet Product Guide for further details on this component.

## Design Entry Method

Instantiation	No
Inference	No
IP and IP Integrator Catalog	Recommended

## For More Information

- See the *UltraScale Architecture Integrated Block for 100G Ethernet v1.7 Product Guide (PG165)*.
- See the [UltraScale Programmable Devices Documentation](#).



# CMACE4

Primitive: 100G MAC Block

PRIMITIVE\_GROUP: ADVANCED

PRIMITIVE\_SUBGROUP: MAC

Families: UltraScale+

## Introduction

The 100G MAC Block provides a high-performance, low latency 100G Ethernet port that allows for a wide range of user customization and statistics gathering. It supports 1588 time stamping for one step and two step. This element is not intended to be instantiated, used, or modified outside of Xilinx generated IP. The block is designed to be integrated with GTs and device clocking resources using fabric interconnect. See the UltraScale+ Architecture Integrated Block for 100G Ethernet Product Guide for further details on this component.

## Design Entry Method

Instantiation	No
Inference	No
IP and IP Integrator Catalog	Recommended

## For More Information

- See the *UltraScale Architecture Integrated Block for 100G Ethernet v1.7 Product Guide (PG165)*.
- See the [UltraScale Programmable Devices Documentation](#).

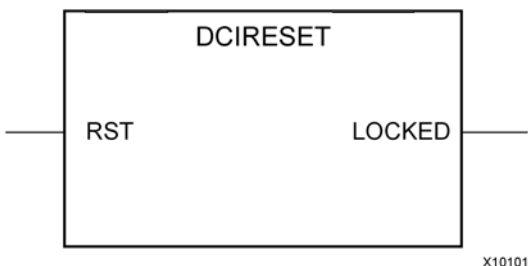
## DCIRESET

Primitive: Digitally Controlled Impedance Reset Component

PRIMITIVE\_GROUP: I/O

PRIMITIVE\_SUBGROUP: DCI\_RESET

Families: UltraScale, UltraScale+



### Introduction

This design element is used to reset the Digitally Controlled Impedance (DCI) state machine after configuration has been completed. By toggling the RST input to the DCIRESET primitive while the device is operating, the DCI state-machine is reset and both phases of impedance adjustment proceed in succession. All I/Os using DCI will be unavailable until the LOCKED output from the DCIRESET block is asserted.

### Port Descriptions

Port	Direction	Width	Function
LOCKED	Output	1	DCI state-machine LOCK status output. When Low, DCI I/O impedance is being calibrated and DCI I/Os are unavailable. Upon a Low-to-High assertion, DCI I/Os are available for use.
RST	Input	1	Active-High asynchronous reset input to DCI state-machine. After RST is asserted, I/Os utilizing DCI will be unavailable until LOCKED is asserted.

### Design Entry Method

Instantiation	Recommended
Inference	No
IP and IP Integrator Catalog	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
--           UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

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
//           UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

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 *UltraScale Architecture SelectIO Resources User Guide* ([UG571](#)).
- See the [UltraScale Programmable Devices Documentation](#).

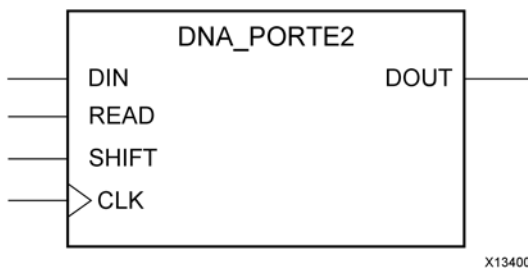
# DNA\_PORTE2

Primitive: Device DNA Access Port

PRIMITIVE\_GROUP: CONFIGURATION

PRIMITIVE\_SUBGROUP: DNA

Families: UltraScale, UltraScale+



## Introduction

The DNA\_PORT allows access to a dedicated shift register that can be loaded with the Device DNA data bits (factory-programmed, read-only unique ID) for a given UltraScale™ 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 device 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 96-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 96-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 zeroes 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
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
SIM_DNA_VALUE	HEX	96'h00000000000000 00000000000000 to 96'hffffffffffffffffffd	All zeroes	Specifies a sample 96-bit DNA value for simulation

## VHDL Instantiation Template

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- DNA_PORTE2: Device DNA Access Port
--           UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

DNA_PORTE2_inst : DNA_PORTE2
generic map (
    SIM_DNA_VALUE => X"00000000000000000000000000000000" -- Specifies a sample 96-bit DNA value for simulation
)
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_PORTE2_inst instantiation
    
```

## Verilog Instantiation Template

```

// DNA_PORTE2: Device DNA Access Port
//           UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

DNA_PORTE2 #(
    .SIM_DNA_VALUE(96'h00000000000000000000000000000000) // Specifies a sample 96-bit DNA value for simulation
)
DNA_PORTE2_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_PORTE2_inst instantiation
    
```

## For More Information

- See the *UltraScale Architecture Configuration User Guide* ([UG570](#)).
- See the [UltraScale Programmable Devices Documentation](#).

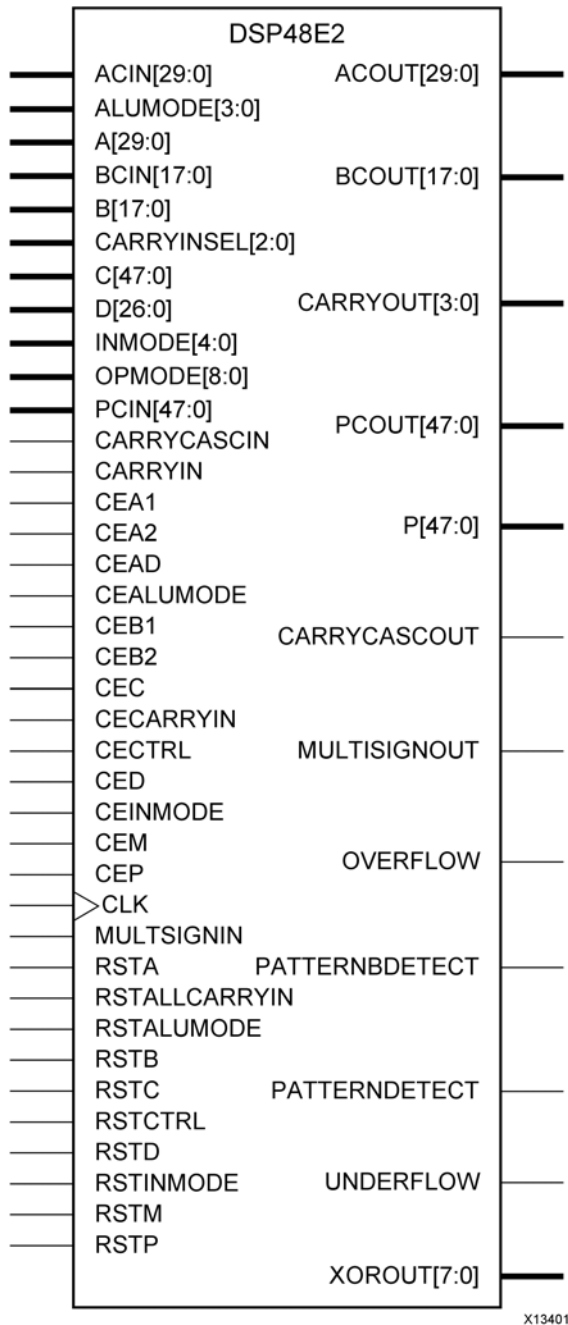
# DSP48E2

Primitive: 48-bit Multi-Functional Arithmetic Block

PRIMITIVE\_GROUP: ARITHMETIC

PRIMITIVE\_SUBGROUP: DSP

Families: UltraScale, UltraScale+



## Introduction

This design element is a versatile, scalable, integrated block 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, subtraction, accumulation, shifting, logical operations and pattern detection.

## Port Descriptions

Port	Direction	Width	Function
XOROUT<7:0>	Output	8	Data output from Wide XOR logic function
<b>Cascade:</b> Cascade Ports			
ACIN<29:0>	Input	30	Cascaded data input from ACOU of previous DSP48E2 (multiplexed with A). If not used, tie port to all zeroes.
ACOU<29:0>	Output	30	Cascaded data output to ACIN of next DSP48E2. If not used, leave unconnected.
BCIN<17:0>	Input	18	Cascaded data input from BCOU of previous DSP48E2 (multiplexed with B). If not used, tie port to all zeroes.
BCOU<17:0>	Output	18	Cascaded data output to BCIN of next DSP48E2. If not used, leave unconnected.
CARRYCASCIN	Input	1	Cascaded carry input from CARRYCASCOU of previous DSP48E2.
CARRYCASCOU	Output	1	Cascaded carry output to CARRYCASCIN of next DSP48E2. This signal is internally fed back into the CARRYINSEL multiplexer input of the same DSP48E2.
MULTSIGNIN	Input	1	Sign of the multiplied result from the previous DSP48E2 for MACC extension. Connect to the MULTSIGNOUT of another DSP block, or tie to ground if not used.
MULTSIGNOUT	Output	1	Sign of the multiplied result cascaded to the next DSP48E2 for MACC extension. Connect to the MULTSIGNIN of another DSP block, or tie to ground if not used.
PCIN<47:0>	Input	48	Cascaded data input from PCOU of previous DSP48E2 to adder. If used, connect to PCOU of upstream cascaded DSP48E2. If not used, tie port to all zeroes.
PCOU<47:0>	Output	48	Cascaded data output to PCIN of next DSP48E2. If used, connect to PCIN of downstream cascaded DSP48E2. If not used, leave unconnected.
<b>Control:</b> Control Inputs/Status Bits			
ALUMODE<3:0>	Input	4	Controls the selection of the logic function in the DSP48E2.
CARRYINSEL<2:0>	Input	3	Selects the carry source: 0 1 1 - PCIN[47] - Rounding PCIN (round towards zero) 1 0 0 - CARRYCASCOU - For larger add/sub/acc (sequential operation via internal feedback). Must select with PREG=1 1 0 1 - ~P[47] - Rounding P (round towards infinity). Must select with PREG=1 1 1 0 - A[24] - XNOR B[17] Rounding A x B 1 1 1 - P[47] - For rounding P (round towards zero). Must select with PREG=1.



Port	Direction	Width	Function
CLK	Input	1	This port is the DSP48E2 input clock, common to all internal registers and flip-flops.
INMODE<4:0>	Input	5	These five control bits select the functionality of the pre-adder, the A, B, and D inputs, and the input registers. These bits should be tied to all zeroes if not used.
OPMODE<8:0>	Input	9	Controls the input to the W, X, Y, and Z multiplexers in the DSP48E2 dictating the operation or function of the component.
OVERFLOW	Output	1	Active-High overflow indicator when used with the appropriate setting of the pattern detector and PREG=1.
PATTERNBDETECT	Output	1	Active-High match indicator between P[47:0] and the pattern bar.
PATTERNDETECT	Output	1	Active-High match indicator between P[47:0] and the pattern gated by the MASK. Result arrives on the same cycle as P.
UNDERFLOW	Output	1	Active-High underflow indicator when used with the appropriate setting of the pattern detector and PREG=1.
<b>Data: Data Ports</b>			
A<29:0>	Input	30	Data input for pre-adder, multiplier, adder/subtractor/accumulator, ALU or concatenation operations. When used with the multiplier or pre-adder, 27 bits of data (A[26:0]) is used and upper bits (A[29:27]) are unused and should be tied High. When using the internal adder/subtractor/accumulator or ALU circuit, all 30 bits are used (A[29:0]). When used in concatenation mode, all 30 bits are used and this constitutes the MSB (upper) bits of the concatenated vector. If this port is not used, tie all bits High.
B<17:0>	Input	18	The B input of the multiplier. B[17:0] are the least significant bits (LSBs) of the A:B concatenated input to the second-stage adder/subtractor or logic function. If this port is not used, tie all bits High.
C<47:0>	Input	48	Data input to the second-stage adder/subtractor, pattern detector, or logic function. If this port is not used, tie all bits High.
CARRYIN	Input	1	Carry input from the device logic.
CARRYOUT<3:0>	Output	4	4-bit carry output from each 12-bit field of the accumulate/adder/logic unit. Normal 48-bit operation uses only CARRYOUT3. SIMD operation can use four carry out bits (CARRYOUT[3:0]).
D<26:0>	Input	27	27-bit data input to the pre-adder or alternative input to the multiplier. The pre-adder implements D + A as determined by the INMODE3 signal. If this port is not used, tie all bits High.
P<47:0>	Output	48	Data output from second stage adder/subtractor or logic function.
<b>Reset/Clock Enable: Reset/Clock Enable Inputs</b>			
CEAD	Input	1	Active-High, clock enable for the pre-adder output AD pipeline register. Tie to logic one if not used and ADREG=1. Tie to logic zero if ADREG=0.

Port	Direction	Width	Function
CEALUMODE	Input	1	Active-High, clock enable for ALUMODE (control inputs) registers (ALUMODEREG=1). Tie to logic one if not used.
CEA1	Input	1	Active-High, clock enable for the first A (input) register. This port is only used if AREG=2 or INMODE0 = 1. 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 the A port is not used, tie Low.
CEA2	Input	1	Active-High, clock enable for the second A (input) register. When two registers are used, this is the second sequentially. When one register is used (AREG=1), CEA2 is the clock enable. If the A port is not used, tie Low.
CEB1	Input	1	Active-High, clock enable for the first B (input) register. When two registers are used, this is the first sequentially. When Dynamic AB Access is used, this clock enable is applied for INMODE[4]=1. If the B port is not used, tie Low.
CEB2	Input	1	Active-High, clock enable for the second B (input) register. This port is only used if BREG=1 or 2. 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. When one register is used (BREG=1), CEB2 is the clock enable.
CEC	Input	1	Active-High, clock enable for the C (input) register (CREG=1). If the C port is not used, tie Low.
CECARRYIN	Input	1	Active-High, clock enable for the CARRYIN (input from fabric) register (CARRYINREG=1). Tie to logic one if not used.
CECTRL	Input	1	Active-High, clock enable for the OPMODE and CARRYINSEL (control inputs) registers (OPMODEREG=1 or CARRYINSELREG=1). Tie to logic one if not used.
CED	Input	1	Active-High, clock enable for the D (input) registers (DREG=1). If the D port is not used, tie Low.
CEINMODE	Input	1	Active-High, clock enable for the INMODE control input registers (INMODEREG=1). Tie to logic one if not used.
CEM	Input	1	Active-High, Clock enable for the post-multiply M (pipeline) register and the internal multiply round CARRYIN register (MREG=1). Tie to logic one if not used.
CEP	Input	1	Active-High, clock enable for the P (output) register (PREG=1). Tie to logic one if not used.
RSTA	Input	1	Synchronous reset for both A (input) registers (AREG=1 or 2). Polarity is determined by the IS_RSTA_INVERTED attribute. Tie to logic zero if A port is not used.
RSTALLCARRYIN	Input	1	Synchronous reset for the Carry (internal path) and the CARRYIN registers (CARRYINREG=1). Polarity is determined by the IS_RSTALLCARRYIN_INVERTED attribute. Tie to logic zero if not used.
RSTALUMODE	Input	1	Synchronous Reset for ALUMODE (control inputs) registers (ALUMODEREG=1). Polarity is determined by the IS_RSTALUMODE_INVERTED attribute. Tie to logic zero if not used.

Port	Direction	Width	Function
RSTB	Input	1	Synchronous Reset for both B (input) registers (BREG=1 or 2). Polarity is determined by the IS_RSTB_INVERTED attribute. Tie to logic zero if B port is not used.
RSTC	Input	1	Synchronous reset for the C (input) registers (CREG=1). Polarity is determined by the IS_RSTC_INVERTED attribute. Tie to logic zero if C port is not used.
RSTCTRL	Input	1	Synchronous reset for OPMODE and CARRYINSEL (control inputs) registers (OPMODEREG=1 and/or CARRYINSELREG=1). Polarity is determined by the IS_RSTCTRL_INVERTED attribute. Tie to logic zero if not used.
RSTD	Input	1	Synchronous reset for the D (input) register and for the pre-adder (output) AD pipeline register (DREG=1 and/or ADREG=1). Polarity is determined by the IS_RSTD_INVERTED attribute. Tie to logic zero if B port is not used.
RSTINMODE	Input	1	Synchronous reset for the INMODE (control input) registers (INMODEREG=1). Polarity is determined by the IS_RSTINMODE_INVERTED attribute. Tie to logic zero if not used.
RSTM	Input	1	Synchronous reset for the M (pipeline) registers (MREG=1). Polarity is determined by the IS_RSTM_INVERTED attribute. Tie to logic zero if not used.
RSTP	Input	1	Synchronous reset for the P (output) registers (PREG=1). Polarity is determined by the IS_RSTP_INVERTED attribute. Tie to logic zero if not used.

## Design Entry Method

Instantiation	Yes
Inference	Recommended
IP and IP Integrator Catalog	Yes

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
<b>Feature Control Attributes:</b> Specifies how to use a given input data port (i.e. from general fabric, "DIRECT", or from another DSP48E2, "CASCADE").				
A_INPUT	STRING	"DIRECT", "CASCADE"	"DIRECT"	Selects the input to the A port between parallel input ("DIRECT") or the cascaded input from the previous DSP48E2 ("CASCADE").
AMULTSEL	STRING	"A", "AD"	"A"	Selects the input to the 27-bit A input of the multiplier
B_INPUT	STRING	"DIRECT", "CASCADE"	"DIRECT"	Selects the input to the B port between parallel input ("DIRECT") or the cascaded input from the previous DSP48E2 ("CASCADE").

Attribute	Type	Allowed Values	Default	Description
BMULTSEL	STRING	"B", "AD"	"B"	Selects the input to the 18-bit B input of the multiplier.
PREADDINSEL	STRING	"A", "B"	"A"	Selects the input to be added with D in the pre-adder
RND	HEX	Any 48-bit HEX value	All zeroes	Rounding Constant into the WMUX.
USE_MULT	STRING	"MULTIPLY", "DYNAMIC", "NONE"	"MULTIPLY"	Selects usage of the multiplier. <ul style="list-style-type: none"> <li>"MULTIPLY" - Multiplier is being used.</li> <li>"NONE" - Saves power when using only the Adder/Logic Unit.</li> <li>"DYNAMIC" - Indicates that the switching between A*B and A:B operations on the fly and therefore needs to get the worst-case timing of the two paths.</li> </ul>
USE_SIMD	STRING	"ONE48", "FOUR12", "TWO24"	"ONE48"	Selects the mode of operation for the adder/subtractor. The attribute setting can be one 48-bit adder mode ("ONE48"), two 24-bit adder mode ("TWO24"), or four 12-bit adder mode ("FOUR12"). Typical Multiply-Add operations are supported when the mode is set to "ONE48". When "TWO24" or "FOUR12" mode is selected, the multiplier must not be used, and USE_MULT must be set to "NONE".
USE_WIDEXOR	STRING	"FALSE", "TRUE"	"FALSE"	Determines whether the Wide XOR is used or not
XORSIMD	STRING	"XOR24_48_96", "XOR12"	"XOR24_48_96"	Selects the mode of operation for the Wide XOR. The attribute setting can be one 96-bit, two 48-bit four 24-bit XOR modes (XOR24_48_96), or eight 12-bit XOR mode (XOR12).
<b>Pattern Detector Attributes:</b> Pattern Detection Configuration/Specification				

Attribute	Type	Allowed Values	Default	Description
AUTORESET_PATDET	STRING	"NO_RESET", "RESET_MATCH", "RESET_NOT_MATCH"	"NO_RESET"	<p>Automatically resets the P Register (accumulated value or counter value) on the next clock cycle, if a pattern detect event has occurred on this clock cycle. The "RESET_MATCH" and "RESET_NOT_MATCH" settings distinguish between whether the DSP48E2 should cause an auto reset of the P Register on the next cycle:</p> <ul style="list-style-type: none"> <li>if the pattern is matched, or</li> <li>whenever the pattern is not matched on the current cycle but was matched on the previous clock cycle</li> </ul>
AUTORESET_PRIORITY	STRING	"RESET", "CEP"	"RESET"	When using the AUTORESET_PATDET feature, defines priority of AUTORESET vs. clock enable (CEP).
MASK	HEX	Any 48-bit HEX value	48'h3fffffff	This 48-bit value is used to mask out certain bits during a pattern detection. When a MASK bit is set to 1, the corresponding pattern bit is ignored. When a MASK bit is set to 0, the pattern bit is compared.
PATTERN	HEX	Any 48-bit HEX value	All zeroes	This 48-bit value is used in the pattern detector.
SEL_MASK	STRING	"MASK", "C", "ROUNDING_MODE1", "ROUNDING_MODE2"	"MASK"	Selects the mask to be used for the pattern detector. The C and MASK settings 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 off of the optionally registered C port. These rounding modes can be used to implement convergent rounding in the DSP48E2 using the pattern detector.
SEL_PATTERN	STRING	"PATTERN", "C"	"PATTERN"	Selects the input source for the pattern field. The input source can be a 48-bit dynamic C input or a 48-bit static PATTERN attribute field.

Attribute	Type	Allowed Values	Default	Description
USE_PATTERN_DETECT	STRING	"NO_PATDET", "PATDET"	"NO_PATDET"	Selects whether the pattern detector and related features are used ("PATDET") or not used ("NO_PATDET"). This attribute is used for speed specification and Simulation Model purposes only.
<p><b>Programmable Inversion Attributes:</b> Specifies whether or not to use the optional inversions on specific pins for this component to change the active polarity of the pin function. When set to 1 on a clock pin (CLK), this component clocks on the negative edge. When set to 1 on other pins, it changes the function to behave active-Low rather than active-High. For pins that are buses, the bit-width of this attribute should match that of the bit-width of the associated pins and a binary value specifies which inverters to use and which to bypass. If an external inverter is specified on one of these associated pins, the Vivado Design Suite will automatically set this attribute during the opt_design stage so that additional logic is not necessary for changing the input polarity.</p>				
IS_ALUMODE_INVERTED	BINARY	4'b0000 to 4'b1111	4'b0000	Specifies whether or not to use the optional inversions on the individual ALUMODE pins of this component.
IS_CARRYIN_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the CARRYIN pin of this component.
IS_CLK_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the CLK pin of this component.
IS_INMODE_INVERTED	BINARY	5'b00000 to 5'b11111	5'b00000	Specifies whether or not to use the optional inversions on the individual INMODE pins of this component.
IS_OPMODE_INVERTED	BINARY	9'b000000000 to 9'b111111111	9'b000000000	Specifies whether or not to use the optional inversions on the individual OPMODE pins of this component.
IS_RSTA_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the RSTA pin of this component.
IS_RSTALLCARRYIN_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the RSTALLCARRYIN pin of this component.
IS_RSTALUMODE_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the RSTALUMODE pin of this component.
IS_RSTB_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the RSTB pin of this component.
IS_RSTC_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the RSTC pin of this component.

Attribute	Type	Allowed Values	Default	Description
IS_RSTCTRL_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the RSTCTRL pin of this component.
IS_RSTD_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the RSTD pin of this component.
IS_RSTINMODE_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the RSTINMODE pin of this component.
IS_RSTM_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the RSTM pin of this component.
IS_RSTP_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the RSTP pin of this component.
<b>Register Control Attributes:</b> Pipeline Register Configuration/Specification				
ACASCREG	DECIMAL	1, 0, 2	1	In conjunction with AREG, selects the number of A input registers on the A cascade path, ACOUT. This attribute must be equal to or one less than the AREG value: <ul style="list-style-type: none"> <li>• AREG=0: ACASCREG must be 0</li> <li>• AREG=1: ACASCREG must be 1</li> <li>• AREG=2: ACASCREG can be 1 or 2</li> </ul>
ADREG	DECIMAL	1, 0	1	Selects the number of pre-adder pipeline registers.
ALUMODEREG	DECIMAL	1, 0	1	Selects the number of ALUMODE input registers.
AREG	DECIMAL	1, 0, 2	1	Selects the number of A input pipeline registers. If A port is not in use, set to 1.
BCASCREG	DECIMAL	1, 0, 2	1	In conjunction with BREG, selects the number of B input registers on the B cascade path, BCOUT. This attribute must be equal to or one less than the BREG value: <ul style="list-style-type: none"> <li>• BREG=0: BCASCREG must be 0</li> <li>• BREG=1: BCASCREG must be 1</li> <li>• BREG=2: BCASCREG can be 1 or 2</li> </ul>

Attribute	Type	Allowed Values	Default	Description
BREG	DECIMAL	1, 0, 2	1	Selects the number of B input registers. If B port is not in use, set to 1.
CARRYINREG	DECIMAL	1, 0	1	Selects the number of CARRYIN input registers.
CARRYINSELREG	DECIMAL	1, 0	1	Selects the number of CARRYINSEL input registers.
CREG	DECIMAL	1, 0	1	Selects the number of C input registers. If C port is not in use, set to 1.
DREG	DECIMAL	1, 0	1	Selects the number of D input registers. If D port is not in use, set to 1.
INMODEREG	DECIMAL	1, 0	1	Selects the number of INMODE input registers.
MREG	DECIMAL	1, 0	1	Selects the number of multiplier output (M) pipeline register stages.
OPMODEREG	DECIMAL	1, 0	1	Selects the number of OPMODE input registers.
PREG	DECIMAL	1, 0	1	Selects the number of P output registers. The registered outputs will include CARRYOUT, CARRYCASCOU, MULTSIGNOUT, PATTERNB_DETECT, PATTERN_DETECT, and PCOUT.

## VHDL Instantiation Template

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- DSP48E2: 48-bit Multi-Functional Arithmetic Block
-- UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

DSP48E2_inst : DSP48E2
generic map (
  -- Feature Control Attributes: Data Path Selection
  AMULTSEL => "A",           -- Selects A input to multiplier (A, AD)
  A_INPUT => "DIRECT",      -- Selects A input source, "DIRECT" (A port) or "CASCADE" (ACIN port)
  BMULTSEL => "B",           -- Selects B input to multiplier (AD, B)
  B_INPUT => "DIRECT",      -- Selects B input source, "DIRECT" (B port) or "CASCADE" (BCIN port)
  PREADDINSEL => "A",       -- Selects input to pre-adder (A, B)
  RND => X"000000000000",    -- Rounding Constant
  USE_MULT => "MULTIPLY",    -- Select multiplier usage (DYNAMIC, MULTIPLY, NONE)
  USE_SIMD => "ONE48",       -- SIMD selection (FOUR12, ONE48, TWO24)
  USE_WIDEXOR => "FALSE",    -- Use the Wide XOR function (FALSE, TRUE)
  XORSIMD => "XOR24_48_96", -- Mode of operation for the Wide XOR (XOR12, XOR24_48_96)
  -- Pattern Detector Attributes: Pattern Detection Configuration
  AUTORESET_PATDET => "NO_RESET", -- NO_RESET, RESET_MATCH, RESET_NOT_MATCH

```



```

AUTORESET_PRIORITY => "RESET",      -- Priority of AUTORESET vs. CEP (CEP, RESET).
MASK => X"3fffffff",                -- 48-bit mask value for pattern detect (l=ignore)
PATTERN => X"000000000000",         -- 48-bit pattern match for pattern detect
SEL_MASK => "MASK",                 -- C, MASK, ROUNDING_MODEL, ROUNDING_MODE2
SEL_PATTERN => "PATTERN",           -- Select pattern value (C, PATTERN)
USE_PATTERN_DETECT => "NO_PATDET",  -- Enable pattern detect (NO_PATDET, PATDET)
-- Programmable Inversion Attributes: Specifies built-in programmable inversion on specific pins
IS_ALUMODE_INVERTED => "0000",     -- Optional inversion for ALUMODE
IS_CARRYIN_INVERTED => '0',        -- Optional inversion for CARRYIN
IS_CLK_INVERTED => '0',            -- Optional inversion for CLK
IS_INMODE_INVERTED => "00000",     -- Optional inversion for INMODE
IS_OPMODE_INVERTED => "000000000", -- Optional inversion for OPMODE
IS_RSTALLCARRYIN_INVERTED => '0',  -- Optional inversion for RSTALLCARRYIN
IS_RSTALUMODE_INVERTED => '0',     -- Optional inversion for RSTALUMODE
IS_RSTA_INVERTED => '0',           -- Optional inversion for RSTA
IS_RSTB_INVERTED => '0',          -- Optional inversion for RSTB
IS_RSTCTRL_INVERTED => '0',       -- Optional inversion for RSTCTRL
IS_RSTC_INVERTED => '0',          -- Optional inversion for RSTC
IS_RSTD_INVERTED => '0',          -- Optional inversion for RSTD
IS_RSTINMODE_INVERTED => '0',     -- Optional inversion for RSTINMODE
IS_RSTM_INVERTED => '0',          -- Optional inversion for RSTM
IS_RSTP_INVERTED => '0',          -- Optional inversion for RSTP
-- Register Control Attributes: Pipeline Register Configuration
ACASCREG => 1,                      -- Number of pipeline stages between A/ACIN and ACOUT (0-2)
ADREG => 1,                          -- Pipeline stages for pre-adder (0-1)
ALUMODEREG => 1,                     -- Pipeline stages for ALUMODE (0-1)
AREG => 1,                            -- Pipeline stages for A (0-2)
BCASCREG => 1,                       -- Number of pipeline stages between B/BCIN and BCOUT (0-2)
BREG => 1,                            -- Pipeline stages for B (0-2)
CARRYINREG => 1,                     -- Pipeline stages for CARRYIN (0-1)
CARRYINSELREG => 1,                  -- Pipeline stages for CARRYINSEL (0-1)
CREG => 1,                            -- Pipeline stages for C (0-1)
DREG => 1,                            -- Pipeline stages for D (0-1)
INMODEREG => 1,                      -- Pipeline stages for INMODE (0-1)
MREG => 1,                            -- Multiplier pipeline stages (0-1)
OPMODEREG => 1,                      -- Pipeline stages for OPMODE (0-1)
PREG => 1                             -- Number of pipeline stages for P (0-1)
)
port map (
-- Cascade outputs: Cascade Ports
ACOUT => ACOUT,                       -- 30-bit output: A port cascade
BCOUT => BCOUT,                       -- 18-bit output: B cascade
CARRYCASCOUT => CARRYCASCOUT,         -- 1-bit output: Cascade carry
MULTSIGNOUT => MULTSIGNOUT,          -- 1-bit output: Multiplier sign cascade
PCOUT => PCOUT,                       -- 48-bit output: Cascade output
-- Control outputs: Control Inputs/Status Bits
OVERFLOW => OVERFLOW,                 -- 1-bit output: Overflow in add/acc
PATTERNBDETECT => PATTERNBDETECT,     -- 1-bit output: Pattern bar detect
PATTERNDETECT => PATTERNDETECT,       -- 1-bit output: Pattern detect
UNDERFLOW => UNDERFLOW,              -- 1-bit output: Underflow in add/acc
-- Data outputs: Data Ports
CARRYOUT => CARRYOUT,                 -- 4-bit output: Carry
P => P,                               -- 48-bit output: Primary data
XOROUT => XOROUT,                     -- 8-bit output: XOR data
-- Cascade inputs: Cascade Ports
ACIN => ACIN,                         -- 30-bit input: A cascade data
BCIN => BCIN,                         -- 18-bit input: B cascade
CARRYCASCIN => CARRYCASCIN,           -- 1-bit input: Cascade carry
MULTSIGNIN => MULTSIGNIN,             -- 1-bit input: Multiplier sign cascade
PCIN => PCIN,                         -- 48-bit input: P cascade
-- Control inputs: Control Inputs/Status Bits
ALUMODE => ALUMODE,                   -- 4-bit input: ALU control
CARRYINSEL => CARRYINSEL,             -- 3-bit input: Carry select
CLK => CLK,                           -- 1-bit input: Clock
INMODE => INMODE,                     -- 5-bit input: INMODE control
OPMODE => OPMODE,                     -- 9-bit input: Operation mode
-- Data inputs: Data Ports
A => A,                               -- 30-bit input: A data
B => B,                               -- 18-bit input: B data
C => C,                               -- 48-bit input: C data
CARRYIN => CARRYIN,                  -- 1-bit input: Carry-in
D => D,                               -- 27-bit input: D data

```

```

-- Reset/Clock Enable inputs: Reset/Clock Enable Inputs
CEA1 => CEA1,           -- 1-bit input: Clock enable for 1st stage AREG
CEA2 => CEA2,           -- 1-bit input: Clock enable for 2nd stage AREG
CEAD => CEAD,           -- 1-bit input: Clock enable for ADREG
CEALUMODE => CEALUMODE, -- 1-bit input: Clock enable for ALUMODE
CEB1 => CEB1,           -- 1-bit input: Clock enable for 1st stage BREG
CEB2 => CEB2,           -- 1-bit input: Clock enable for 2nd stage BREG
CEC => CEC,             -- 1-bit input: Clock enable for CREG
CECARRYIN => CECARRYIN, -- 1-bit input: Clock enable for CARRYINREG
CECTRL => CECTRL,       -- 1-bit input: Clock enable for OPMODEREG and CARRYINSELREG
CED => CED,             -- 1-bit input: Clock enable for DREG
CEINMODE => CEINMODE,   -- 1-bit input: Clock enable for INMODEREG
CEM => CEM,             -- 1-bit input: Clock enable for MREG
CEP => CEP,             -- 1-bit input: Clock enable for PREG
RSTA => RSTA,           -- 1-bit input: Reset for AREG
RSTALLCARRYIN => RSTALLCARRYIN, -- 1-bit input: Reset for CARRYINREG
RSTALUMODE => RSTALUMODE, -- 1-bit input: Reset for ALUMODEREG
RSTB => RSTB,           -- 1-bit input: Reset for BREG
RSTC => RSTC,           -- 1-bit input: Reset for CREG
RSTCTRL => RSTCTRL,    -- 1-bit input: Reset for OPMODEREG and CARRYINSELREG
RSTD => RSTD,           -- 1-bit input: Reset for DREG and ADREG
RSTINMODE => RSTINMODE, -- 1-bit input: Reset for INMODEREG
RSTM => RSTM,           -- 1-bit input: Reset for MREG
RSTP => RSTP           -- 1-bit input: Reset for PREG
);

-- End of DSP48E2_inst instantiation
    
```

## Verilog Instantiation Template

```

// DSP48E2: 48-bit Multi-Functional Arithmetic Block
// UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

DSP48E2 #(
    // Feature Control Attributes: Data Path Selection
    .AMULTSEL("A"),           // Selects A input to multiplier (A, AD)
    .A_INPUT("DIRECT"),      // Selects A input source, "DIRECT" (A port) or "CASCADE" (ACIN port)
    .BMULTSEL("B"),          // Selects B input to multiplier (AD, B)
    .B_INPUT("DIRECT"),      // Selects B input source, "DIRECT" (B port) or "CASCADE" (BCIN port)
    .PREADDINSEL("A"),       // Selects input to pre-adder (A, B)
    .RND(48'h000000000000),  // Rounding Constant
    .USE_MULT("MULTIPLY"),    // Select multiplier usage (DYNAMIC, MULTIPLY, NONE)
    .USE_SIMD("ONE48"),      // SIMD selection (FOUR12, ONE48, TWO24)
    .USE_WIDEXOR("FALSE"),   // Use the Wide XOR function (FALSE, TRUE)
    .XORSIMD("XOR24_48_96"), // Mode of operation for the Wide XOR (XOR12, XOR24_48_96)
    // Pattern Detector Attributes: Pattern Detection Configuration
    .AUTORESET_PATDET("NO_RESET"), // NO RESET, RESET_MATCH, RESET_NOT_MATCH
    .AUTORESET_PRIORITY("RESET"), // Priority of AUTORESET vs. CEP (CEP, RESET).
    .MASK(48'h3fffffff),        // 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 (C, PATTERN)
    .USE_PATTERN_DETECT("NO_PATDET"), // Enable pattern detect (NO_PATDET, PATDET)
    // Programmable Inversion Attributes: Specifies built-in programmable inversion on specific pins
    .IS_ALUMODE_INVERTED(4'b0000), // Optional inversion for ALUMODE
    .IS_CARRYIN_INVERTED(1'b0),    // Optional inversion for CARRYIN
    .IS_CLK_INVERTED(1'b0),        // Optional inversion for CLK
    .IS_INMODE_INVERTED(5'b000000), // Optional inversion for INMODE
    .IS_OPMODE_INVERTED(9'b00000000), // Optional inversion for OPMODE
    .IS_RSTALLCARRYIN_INVERTED(1'b0), // Optional inversion for RSTALLCARRYIN
    .IS_RSTALUMODE_INVERTED(1'b0), // Optional inversion for RSTALUMODE
    .IS_RSTA_INVERTED(1'b0),       // Optional inversion for RSTA
    .IS_RSTB_INVERTED(1'b0),       // Optional inversion for RSTB
    .IS_RSTCTRL_INVERTED(1'b0),    // Optional inversion for RSTCTRL
    .IS_RSTC_INVERTED(1'b0),       // Optional inversion for RSTC
    .IS_RSTD_INVERTED(1'b0),       // Optional inversion for RSTD
    .IS_RSTINMODE_INVERTED(1'b0), // Optional inversion for RSTINMODE
    .IS_RSTM_INVERTED(1'b0),       // Optional inversion for RSTM
    .IS_RSTP_INVERTED(1'b0),       // Optional inversion for RSTP
    )
    
```

```

// Register Control Attributes: Pipeline Register Configuration
.ACASCREG(1), // Number of pipeline stages between A/ACIN and ACOUT (0-2)
.ADREG(1), // Pipeline stages for pre-adder (0-1)
.ALUMODEREG(1), // Pipeline stages for ALUMODE (0-1)
.AREG(1), // Pipeline stages for A (0-2)
.BCASCREG(1), // Number of pipeline stages between B/BCIN and BCOUT (0-2)
.BREG(1), // Pipeline stages for B (0-2)
.CARRYINREG(1), // Pipeline stages for CARRYIN (0-1)
.CARRYINSELREG(1), // Pipeline stages for CARRYINSEL (0-1)
.CREG(1), // Pipeline stages for C (0-1)
.DREG(1), // Pipeline stages for D (0-1)
.INMODEREG(1), // Pipeline stages for INMODE (0-1)
.MREG(1), // Multiplier pipeline stages (0-1)
.OPMODEREG(1), // Pipeline stages for OPMODE (0-1)
.PREG(1) // Number of pipeline stages for P (0-1)
)
DSP48E2_inst (
// Cascade outputs: Cascade Ports
.ACOUT(ACOUT), // 30-bit output: A port cascade
.BCOUT(BCOUT), // 18-bit output: B cascade
.CARRYCASCOUT(CARRYCASCOUT), // 1-bit output: Cascade carry
.MULTSIGNOUT(MULTSIGNOUT), // 1-bit output: Multiplier sign cascade
.PCOUT(PCOUT), // 48-bit output: Cascade output
// Control outputs: Control Inputs/Status Bits
.OVERFLOW(OVERFLOW), // 1-bit output: Overflow in add/acc
.PATTERNBDETECT(PATTERNBDETECT), // 1-bit output: Pattern bar detect
.PATTERNDETECT(PATTERNDETECT), // 1-bit output: Pattern detect
.UNDERFLOW(UNDERFLOW), // 1-bit output: Underflow in add/acc
// Data outputs: Data Ports
.CARRYOUT(CARRYOUT), // 4-bit output: Carry
.P(P), // 48-bit output: Primary data
.XOROUT(XOROUT), // 8-bit output: XOR data
// Cascade inputs: Cascade Ports
.ACIN(ACIN), // 30-bit input: A cascade data
.BCIN(BCIN), // 18-bit input: B cascade
.CARRYCASCIN(CARRYCASCIN), // 1-bit input: Cascade carry
.MULTSIGNIN(MULTSIGNIN), // 1-bit input: Multiplier sign cascade
.PCIN(PCIN), // 48-bit input: P cascade
// Control inputs: Control Inputs/Status Bits
.ALUMODE(ALUMODE), // 4-bit input: ALU control
.CARRYINSEL(CARRYINSEL), // 3-bit input: Carry select
.CLK(CLK), // 1-bit input: Clock
.INMODE(INMODE), // 5-bit input: INMODE control
.OPMODE(OPMODE), // 9-bit input: Operation mode
// Data inputs: Data Ports
.A(A), // 30-bit input: A data
.B(B), // 18-bit input: B data
.C(C), // 48-bit input: C data
.CARRYIN(CARRYIN), // 1-bit input: Carry-in
.D(D), // 27-bit input: D data
// Reset/Clock Enable inputs: Reset/Clock Enable Inputs
.CEA1(CEA1), // 1-bit input: Clock enable for 1st stage AREG
.CEA2(CEA2), // 1-bit input: Clock enable for 2nd stage AREG
.CEAD(CEAD), // 1-bit input: Clock enable for ADREG
.CEALUMODE(CEALUMODE), // 1-bit input: Clock enable for ALUMODE
.CEB1(CEB1), // 1-bit input: Clock enable for 1st stage BREG
.CEB2(CEB2), // 1-bit input: Clock enable for 2nd stage BREG
.CEC(CEC), // 1-bit input: Clock enable for CREG
.CECARRYIN(CECARRYIN), // 1-bit input: Clock enable for CARRYINREG
.CECTRL(CECTRL), // 1-bit input: Clock enable for OPMODEREG and CARRYINSELREG
.CED(CED), // 1-bit input: Clock enable for DREG
.CEINMODE(CEINMODE), // 1-bit input: Clock enable for INMODEREG
.CEM(CEM), // 1-bit input: Clock enable for MREG
.CEP(CEP), // 1-bit input: Clock enable for PREG
.RSTA(RSTA), // 1-bit input: Reset for AREG
.RSTALLCARRYIN(RSTALLCARRYIN), // 1-bit input: Reset for CARRYINREG
.RSTALUMODE(RSTALUMODE), // 1-bit input: Reset for ALUMODEREG
.RSTB(RSTB), // 1-bit input: Reset for BREG
.RSTC(RSTC), // 1-bit input: Reset for CREG
.RSTCTRL(RSTCTRL), // 1-bit input: Reset for OPMODEREG and CARRYINSELREG
.RSTD(RSTD), // 1-bit input: Reset for DREG and ADREG
.RSTINMODE(RSTINMODE), // 1-bit input: Reset for INMODEREG

```

```
.RSTM(RSTM),           // 1-bit input: Reset for MREG
.RSTP(RSTP)           // 1-bit input: Reset for PREG
);

// End of DSP48E2_inst instantiation
```

## For More Information

- See the *UltraScale Architecture DSP Slice User Guide* ([UG579](#)).
- See the [UltraScale Programmable Devices Documentation](#).

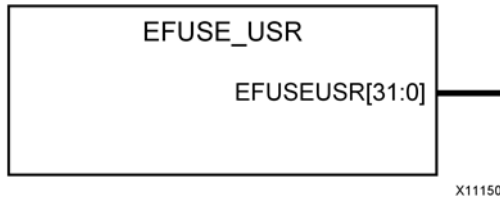
## EFUSE\_USR

Primitive: 32-bit non-volatile design ID

PRIMITIVE\_GROUP: CONFIGURATION

PRIMITIVE\_SUBGROUP: EFUSE

Families: UltraScale, UltraScale+



### Introduction

Provides internal access to the 32 non-volatile, user-programmable eFUSE bits

### Port Descriptions

Port	Direction	Width	Function
EFUSEUSR<31:0>	Output	32	User eFUSE register value output

### Design Entry Method

Instantiation	Recommended
Inference	No
IP and IP Integrator Catalog	No

### Available Attributes

Attribute	Type	Allowed Values	Default	Description
SIM_EFUSE_VALUE	HEX	Any 32-bit HEX value	All zeroes	Value of the 32-bit non-volatile value 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
--           UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

EFUSE_USR_inst : EFUSE_USR
```

```
generic map (  
    SIM_EFUSE_VALUE => X"00000000" -- Value of the 32-bit non-volatile value used in simulation  
)  
port map (  
    EFUSEUSR => EFUSEUSR -- 32-bit output: User eFUSE register value output  
);  
  
-- End of EFUSE_USR_inst instantiation
```

## Verilog Instantiation Template

```
// EFUSE_USR: 32-bit non-volatile design ID  
// UltraScale  
// Xilinx HDL Libraries Guide, version 2016.1  
  
EFUSE_USR #(  
    .SIM_EFUSE_VALUE(32'h00000000) // Value of the 32-bit non-volatile value used in simulation  
)  
EFUSE_USR_inst (  
    .EFUSEUSR(EFUSEUSR) // 32-bit output: User eFUSE register value output  
);  
  
// End of EFUSE_USR_inst instantiation
```

## For More Information

- See the *UltraScale Architecture Configuration User Guide* ([UG570](#)).
- See the [UltraScale Programmable Devices Documentation](#).

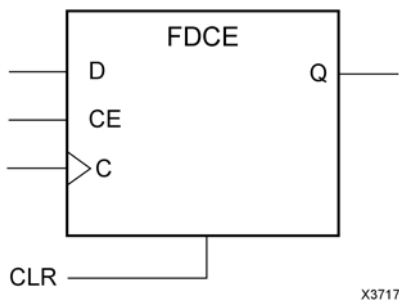
## FDCE

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

PRIMITIVE\_GROUP: REGISTER

PRIMITIVE\_SUBGROUP: SDR

Families: UltraScale, UltraScale+



## 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 not asserted, the data on the data input (D) of this design element is transferred to the corresponding data output (Q) during the clock (C) transition. When CLR is active, 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 initialized when power is applied. When global set/reset (GSR) is active upon power-up or when asserted via the STARTUP block, the value of the INIT attribute is placed on the register's output.

## Logic Table

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

## Port Descriptions

Port	Direction	Width	Function
C	Input	1	Clock input. Polarity is determined by the IS_C_INVERTED attribute.
CE	Input	1	Active-High register clock enable
CLR	Input	1	Asynchronous clear. Polarity is determined by the IS_CLR_INVERTED attribute.
D	Input	1	Data input
Q	Output	1	Data output

## Design Entry Method

Instantiation	Yes
Inference	Recommended
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	BINARY	1'b0, 1'b1	1'b0	Sets the initial value of Q output after configuration or when GSR is asserted via the STARTUP block
<b>Programmable Inversion Attributes:</b> Specifies whether or not to use the optional inversion on specific pins for this component to change the active polarity of the pin function. When set to 1 on a clock pin (C), it creates a negative edge register. When set to 1 on other pins, it changes the function to behave active-Low rather than active-High. If an external inverter is specified on one of these associated pins, the Vivado Design Suite will automatically set this attribute during the opt_design stage so that additional logic is not necessary for changing the input polarity.				
IS_C_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion for the C pin of this component.
IS_CLR_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion for the CLR pin of this component.
IS_D_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion for the D pin of this component. Must be set to 0 unless used as an I/O register.

## VHDL Instantiation Template

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- FDCE: D Flip-Flop with Clock Enable and Asynchronous Clear
-- UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

FDCE_inst : FDCE
generic map (
    INIT => '0',           -- Initial value of register, '0', '1'
    -- Programmable Inversion Attributes: Specifies the use of the built-in programmable inversion
    IS_CLR_INVERTED => '0', -- Optional inversion for CLR
    IS_C_INVERTED => '0',  -- Optional inversion for C
    IS_D_INVERTED => '0'  -- Optional inversion for D
)
port map (
    Q => Q,           -- 1-bit output: Data
    C => C,           -- 1-bit input: Clock
    CE => CE,        -- 1-bit input: Clock enable
    CLR => CLR,      -- 1-bit input: Asynchronous clear
    D => D            -- 1-bit input: Data
);

-- End of FDCE_inst instantiation
    
```



## Verilog Instantiation Template

```
// FDCE: D Flip-Flop with Clock Enable and Asynchronous Clear
//      UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

FDCE #(
    .INIT(1'b0),           // Initial value of register, 1'b0, 1'b1
    // Programmable Inversion Attributes: Specifies the use of the built-in programmable inversion
    .IS_CLR_INVERTED(1'b0), // Optional inversion for CLR
    .IS_C_INVERTED(1'b0),  // Optional inversion for C
    .IS_D_INVERTED(1'b0)  // Optional inversion for D
)
FDCE_inst (
    .Q(Q),      // 1-bit output: Data
    .C(C),      // 1-bit input: Clock
    .CE(CE),    // 1-bit input: Clock enable
    .CLR(CLR),  // 1-bit input: Asynchronous clear
    .D(D)       // 1-bit input: Data
);

// End of FDCE_inst instantiation
```

## For More Information

- See the *UltraScale Architecture Configurable Logic Block User Guide* ([UG574](#)).
- See the [UltraScale Programmable Devices Documentation](#).

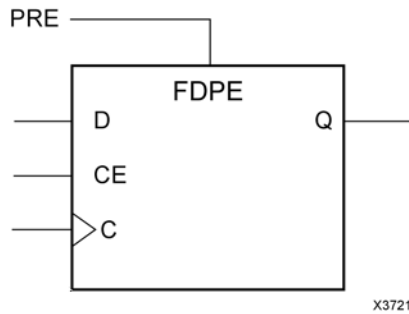
## FDPE

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

PRIMITIVE\_GROUP: REGISTER

PRIMITIVE\_SUBGROUP: SDR

Families: UltraScale, UltraScale+



## Introduction

This design element is a single D-type flip-flop with clock enable and asynchronous preset. When clock enable (CE) is High and asynchronous preset (PRE) is not asserted, the data on the data input (D) of this design element is transferred to the corresponding data output (Q) during the clock (C) transition. When PRE is asserted, it overrides all other inputs and presets the data output (Q) High. When CE is Low, clock transitions are ignored.

This flip-flop is asynchronously initialized when power is applied. When global set/reset (GSR) is active upon power-up or when asserted via the STARTUP block, the value of the INIT attribute is placed on the register's output.

## Logic Table

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

## Port Descriptions

Port	Direction	Width	Function
C	Input	1	Clock input. Polarity is determined by the IS_C_INVERTED attribute.
CE	Input	1	Active-High register clock enable
D	Input	1	Data input
PRE	Input	1	Asynchronous preset. Polarity is determined by the IS_PRE_INVERTED attribute.
Q	Output	1	Data output

## Design Entry Method

Instantiation	Yes
Inference	Recommended
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	BINARY	1'b1, 1'b0	1'b1	Sets the initial value of Q output after configuration or when GSR is asserted via the STARTUP block
<b>Programmable Inversion Attributes:</b> Specifies whether or not to use the optional inversion on specific pins for this component to change the active polarity of the pin function. When set to 1 on a clock pin (C), it creates a negative edge register. When set to 1 on other pins, it changes the function to behave active-Low rather than active-High. If an external inverter is specified on one of these associated pins, the Vivado Design Suite will automatically set this attribute during the opt_design stage so that additional logic is not necessary for changing the input polarity.				
IS_C_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion for the C pin of this component.
IS_D_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion for the D pin of this component. Must be set to 0 unless used as an I/O register.
IS_PRE_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion for the PRE pin of this component.

## VHDL Instantiation Template

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- FDPE: D Flip-Flop with Clock Enable and Asynchronous Preset
-- UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

FDPE_inst : FDPE
generic map (
    INIT => '1',           -- Initial value of register, '0', '1'
    -- Programmable Inversion Attributes: Specifies the use of the built-in programmable inversion
    IS_C_INVERTED => '0',  -- Optional inversion for C
    IS_D_INVERTED => '0',  -- Optional inversion for D
    IS_PRE_INVERTED => '0' -- Optional inversion for PRE
)
port map (
    Q => Q,           -- 1-bit output: Data
    C => C,           -- 1-bit input: Clock
    CE => CE,        -- 1-bit input: Clock enable
    D => D,           -- 1-bit input: Data
    PRE => PRE       -- 1-bit input: Asynchronous preset
);

-- End of FDPE_inst instantiation
    
```

## Verilog Instantiation Template

```
// FDPE: D Flip-Flop with Clock Enable and Asynchronous Preset
//      UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

FDPE #(
    .INIT(1'b1),           // Initial value of register, 1'b0, 1'b1
    // Programmable Inversion Attributes: Specifies the use of the built-in programmable inversion
    .IS_C_INVERTED(1'b0), // Optional inversion for C
    .IS_D_INVERTED(1'b0), // Optional inversion for D
    .IS_PRE_INVERTED(1'b0) // Optional inversion for PRE
)
FDPE_inst (
    .Q(Q),           // 1-bit output: Data
    .C(C),           // 1-bit input: Clock
    .CE(CE),         // 1-bit input: Clock enable
    .D(D),           // 1-bit input: Data
    .PRE(PRE)        // 1-bit input: Asynchronous preset
);

// End of FDPE_inst instantiation
```

## For More Information

- See the *UltraScale Architecture Configurable Logic Block User Guide* ([UG574](#)).
- See the [UltraScale Programmable Devices Documentation](#).

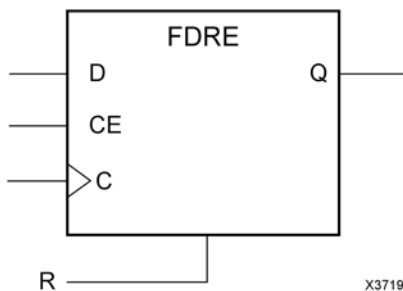
## FDRE

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

PRIMITIVE\_GROUP: REGISTER

PRIMITIVE\_SUBGROUP: SDR

Families: UltraScale, UltraScale+



## Introduction

This design element is a single D-type flip-flop with clock enable and synchronous reset. When clock enable (CE) is High and synchronous reset (R) is not asserted, the data on the data input (D) of this design element is transferred to the corresponding data output (Q) during the clock (C) transition. When R is active, it overrides all other inputs and resets the data output (Q) Low upon the next clock transition. When CE is Low, clock transitions are ignored.

This flip-flop is asynchronously initialized when power is applied. When global set/reset (GSR) is active upon power-up or when asserted via the STARTUP block, the value of the INIT attribute is placed on the register's output.

## Logic Table

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

## Port Descriptions

Port	Direction	Width	Function
C	Input	1	Clock input. Polarity is determined by the IS_C_INVERTED attribute.
CE	Input	1	Active-High register clock enable
D	Input	1	Data input
Q	Output	1	Data output
R	Input	1	Synchronous reset. Polarity is determined by the IS_R_INVERTED attribute.

## Design Entry Method

Instantiation	Yes
Inference	Recommended
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	BINARY	1'b0, 1'b1	1'b0	Sets the initial value of Q output after configuration or when GSR is asserted via the STARTUP block.
<b>Programmable Inversion Attributes:</b> Specifies whether or not to use the optional inversion on specific pins for this component to change the active polarity of the pin function. When set to 1 on a clock pin (C), it creates a negative edge register. When set to 1 on other pins, it changes the function to behave active-Low rather than active-High. If an external inverter is specified on one of these associated pins, the Vivado Design Suite will automatically set this attribute during the opt_design stage so that additional logic is not necessary for changing the input polarity.				
IS_C_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion for the C pin of this component.
IS_D_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion for the D pin of this component. Must be set to 0 unless used as an I/O register.
IS_R_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion for the R pin of this component

## VHDL Instantiation Template

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- FDRE: D Flip-Flop with Clock Enable and Synchronous Reset
-- UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

FDRE_inst : FDRE
generic map (
    INIT => '0',           -- Initial value of register, '0', '1'
    -- Programmable Inversion Attributes: Specifies the use of the built-in programmable inversion
    IS_C_INVERTED => '0', -- Optional inversion for C
    IS_D_INVERTED => '0', -- Optional inversion for D
    IS_R_INVERTED => '0'  -- Optional inversion for R
)
port map (
    Q => Q,    -- 1-bit output: Data
    C => C,    -- 1-bit input: Clock
    CE => CE,  -- 1-bit input: Clock enable
    D => D,    -- 1-bit input: Data
    R => R     -- 1-bit input: Synchronous reset
);

-- End of FDRE_inst instantiation
    
```

## Verilog Instantiation Template

```
// FDRE: D Flip-Flop with Clock Enable and Synchronous Reset
//      UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

FDRE #(
    .INIT(1'b0),           // Initial value of register, 1'b0, 1'b1
    // Programmable Inversion Attributes: Specifies the use of the built-in programmable inversion
    .IS_C_INVERTED(1'b0), // Optional inversion for C
    .IS_D_INVERTED(1'b0), // Optional inversion for D
    .IS_R_INVERTED(1'b0)  // Optional inversion for R
)
FDRE_inst (
    .Q(Q), // 1-bit output: Data
    .C(C), // 1-bit input: Clock
    .CE(CE), // 1-bit input: Clock enable
    .D(D), // 1-bit input: Data
    .R(R) // 1-bit input: Synchronous reset
);

// End of FDRE_inst instantiation
```

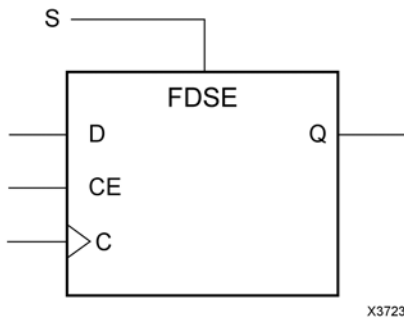
## For More Information

- See the *UltraScale Architecture Configurable Logic Block User Guide* ([UG574](#)).
- See the [UltraScale Programmable Devices Documentation](#).

# FDSE

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

PRIMITIVE\_GROUP: REGISTER  
 PRIMITIVE\_SUBGROUP: SDR  
 Families: UltraScale, UltraScale+



## Introduction

This design element is a single D-type flip-flop with clock enable and synchronous set. When clock enable (CE) is High and synchronous set (S) is not asserted, the data on the data input (D) of this design element is transferred to the corresponding data output (Q) during the clock (C) transition. When S is active, it overrides all other inputs and sets the data output (Q) High upon the next clock transition. When CE is Low, clock transitions are ignored.

This flip-flop is asynchronously initialized when power is applied. When global set/reset (GSR) is active upon power-up or when asserted via the STARTUP block, the value of the INIT attribute is placed on the register's output.

## Logic Table

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

## Port Descriptions

Port	Direction	Width	Function
C	Input	1	Clock input. Polarity is determined by the IS_C_INVERTED attribute.
CE	Input	1	Active-High register clock enable
D	Input	1	Data input
Q	Output	1	Data output
S	Input	1	Synchronous set. Polarity is determined by the IS_S_INVERTED attribute.



## Design Entry Method

Instantiation	Yes
Inference	Recommended
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	BINARY	1'b1, 1'b0	1'b1	Sets the initial value of Q output after configuration or when GSR is asserted via the STARTUP block.
<b>Programmable Inversion Attributes:</b> Specifies whether or not to use the optional inversion on specific pins for this component to change the active polarity of the pin function. When set to 1 on a clock pin (C), it creates a negative edge register. When set to 1 on other pins, it changes the function to behave active-Low rather than active-High. If an external inverter is specified on one of these associated pins, the Vivado Design Suite will automatically set this attribute during the opt_design stage so that additional logic is not necessary for changing the input polarity.				
IS_C_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion for the C pin of this component.
IS_D_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion for the D pin of this component. Must be set to 0 unless used as an I/O register.
IS_S_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion for the S pin of this component

## VHDL Instantiation Template

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- FDSE: D Flip-Flop with Clock Enable and Synchronous Set
-- UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

FDSE_inst : FDSE
generic map (
    INIT => '1',          -- Initial value of register, '0', '1'
    -- Programmable Inversion Attributes: Specifies the use of the built-in programmable inversion
    IS_C_INVERTED => '0', -- Optional inversion for C
    IS_D_INVERTED => '0', -- Optional inversion for D
    IS_S_INVERTED => '0' -- Optional inversion for S
)
port map (
    Q => Q,    -- 1-bit output: Data
    C => C,    -- 1-bit input: Clock
    CE => CE,  -- 1-bit input: Clock enable
    D => D,    -- 1-bit input: Data
    S => S     -- 1-bit input: Synchronous set
);

-- End of FDSE_inst instantiation
    
```

## Verilog Instantiation Template

```
// FDSE: D Flip-Flop with Clock Enable and Synchronous Set
//      UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

FDSE #(
    .INIT(1'b1),           // Initial value of register, 1'b0, 1'b1
    // Programmable Inversion Attributes: Specifies the use of the built-in programmable inversion
    .IS_C_INVERTED(1'b0), // Optional inversion for C
    .IS_D_INVERTED(1'b0), // Optional inversion for D
    .IS_S_INVERTED(1'b0) // Optional inversion for S
)
FDSE_inst (
    .Q(Q), // 1-bit output: Data
    .C(C), // 1-bit input: Clock
    .CE(CE), // 1-bit input: Clock enable
    .D(D), // 1-bit input: Data
    .S(S) // 1-bit input: Synchronous set
);

// End of FDSE_inst instantiation
```

## For More Information

- See the *UltraScale Architecture Configurable Logic Block User Guide* ([UG574](#)).
- See the [UltraScale Programmable Devices Documentation](#).

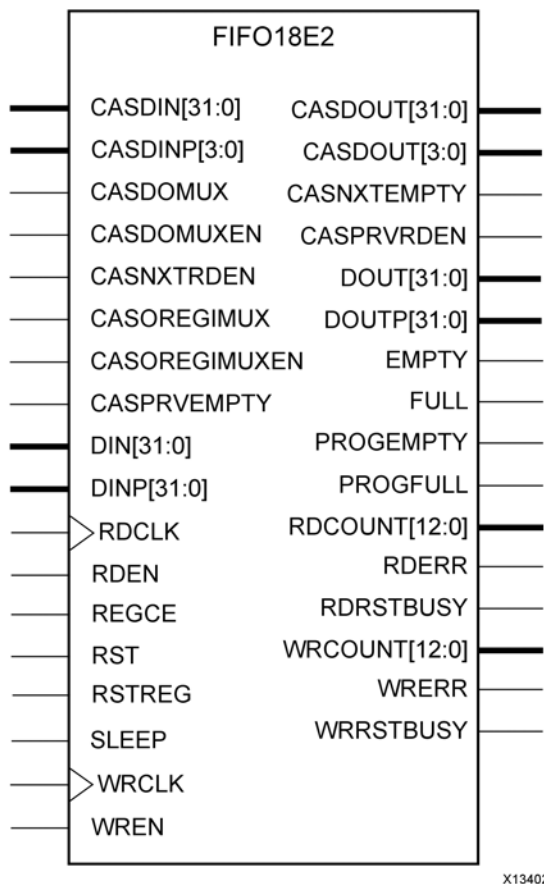
# FIFO18E2

Primitive: 18Kb FIFO (First-In-First-Out) Block RAM Memory

PRIMITIVE\_GROUP: BLOCKRAM

PRIMITIVE\_SUBGROUP: FIFO

Families: UltraScale, UltraScale+



## Introduction

The FIFO18E2 uses dedicated control logic and the 18Kb Block RAM to deliver a configurable First-In-First-Out (FIFO) capability. 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 synchronous or dual-clock (asynchronous) mode, with all associated FIFO flags and status signals. This FIFO also features a cascade capability which lets you chain multiple FIFO18E2 components to form deeper FIFO configurations if desired.

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.

## Port Descriptions

Port	Direction	Width	Function
<b>Cascade Signals:</b> Signals used when cascading more than one FIFO			
CASDIN<31:0>	Input	32	Data input bus from previous CASDOUT bus when cascading FIFOs serially or in parallel to extend depth. Only used when CASCADE_ORDER="MIDDLE", "LAST", or "PARALLEL".
CASDINP<3:0>	Input	4	Parity data input bus from previous CASDOUTP bus when cascading FIFOs serially or in parallel to extend depth. Only used when CASCADE_ORDER="MIDDLE", "LAST", or "PARALLEL".
CASDOMUX	Input	1	D input to flop that drives the select line to the cascade mux on the BRAM outputs. Only used when CASCADE_ORDER="PARALLEL".
CASDOMUXEN	Input	1	EN input to flop that drives the select line to the cascade mux on the BRAM outputs. Only used when CASCADE_ORDER="PARALLEL".
CASDOUT<31:0>	Output	32	Data output bus to next CASDIN bus when cascading FIFOs serially or in parallel to extend depth. Only used when CASCADE_ORDER="FIRST", "MIDDLE", or "PARALLEL".
CASDOUTP<3:0>	Output	4	Parity data output bus to next CASDINP bus when cascading FIFOs serially or in parallel to extend depth. Only used when CASCADE_ORDER="FIRST", "MIDDLE", or "PARALLEL".
CASNXTEMPTY	Output	1	EMPTY input from previous FIFO, used for cascading FIFOs serially to extend depth. Connects to the CASNXTEMPTY of the previous FIFO. Only used when CASCADE_ORDER="MIDDLE" or "LAST".
CASNXRDEN	Input	1	RDEN input from next FIFO, used for cascading FIFOs serially to extend depth. Connects to CASPRVRDEN of the next FIFO. Only used when CASCADE_ORDER="FIRST" or "MIDDLE".
CASOREGIMUX	Input	1	D input to flop that drives the select line to the cascade mux before the output registers. Only used when CASCADE_ORDER="PARALLEL" and REGISTER_MODE="DO_PIPELINED".
CASOREGIMUXEN	Input	1	EN input to flop that drives the select line to the cascade mux before the output registers. Only used when CASCADE_ORDER="PARALLEL" and REGISTER_MODE="DO_PIPELINED".
CASPRVEMPTY	Input	1	EMPTY input from previous FIFO, used for cascading FIFOs serially to extend depth. Connects to the CASNXTEMPTY of the previous FIFO. Only used when CASCADE_ORDER is "MIDDLE" or "LAST".
CASPRVRDEN	Output	1	Control output driving the RDEN input of the previous (PRV) FIFO, used for cascading FIFOs serially to extend depth. Connects to CASNXRDEN of the previous FIFO. Only used when CASCADE_ORDER is "MIDDLE" or "LAST".
<b>Read Control Signals:</b> Read clock, enable and reset input signals			
RDCLK	Input	1	Read clock.
RDEN	Input	1	Active-High read enable.
REGCE	Input	1	Active-High enable for output register stage.
RSTREG	Input	1	Active-High enable for output register reset.
SLEEP	Input	1	Dynamic shut down power saving. If SLEEP is High, the block is in power saving mode. If SLEEP_ASYNC=FALSE, synchronous to RDCLK, otherwise, asynchronous input.
<b>Read Data:</b> Read output data			

Port	Direction	Width	Function
DOUT<31:0>	Output	32	FIFO data output bus. Synchronous to RDCLK.
DOUPT<3:0>	Output	4	FIFO parity output bus. Synchronous to RDCLK.
<b>Status:</b> Flags and other FIFO status outputs			
EMPTY	Output	1	Active-High flag to indicate when the FIFO is empty. Synchronous to RDCLK.
FULL	Output	1	Active-High flag to indicate when the FIFO is full. Synchronous to WRCLK.
PROGEMPTY	Output	1	Programmable flag to indicate the FIFO is almost empty (contains less than or equal to the number of words specified by the PROG_EMPTY_THRESH). Synchronous to RDCLK.
PROGFULL	Output	1	Programmable flag to indicate the FIFO is almost full (contains greater than or equal to the number of words specified by the PROG_FULL_THRESH) Synchronous to WRCLK.
RDCOUNT<12:0>	Output	13	Output of the internal FIFO read pointer, or a count of the number of words in the FIFO. Synchronous to RDCLK. Output value controlled by RDCOUNT_TYPE.
RDERR	Output	1	Indicates that a read operation failed due to FIFO being EMPTY, or FIFO being in a reset condition. Synchronous to RDCLK.
RDRSTBUSY	Output	1	Active-High indicator that the FIFO is currently in a reset state. Synchronous to RDCLK.
WRCOUNT<12:0>	Output	13	Output of the internal FIFO write pointer, or a count of the number of words in the FIFO. Synchronous to WRCLK. Output value controlled by WRCOUNT_TYPE.
WRERR	Output	1	Indicates that a write operation failed due to FIFO being FULL, or FIFO being in a reset condition. Synchronous to WRCLK.
WRRSTBUSY	Output	1	Active-High indicator that the FIFO is currently in a reset state. Synchronous to WRCLK.
<b>Write Control Signals:</b> Write clock and enable input signals			
RST	Input	1	Active-High synchronous reset. Synchronous to WRCLK.
WRCLK	Input	1	Write clock.
WREN	Input	1	Active-High write enable.
<b>Write Data:</b> Write input data			
DIN<31:0>	Input	32	FIFO data input bus. Synchronous to WRCLK.
DINP<3:0>	Input	4	FIFO parity input bus. Synchronous to WRCLK.

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	Recommended

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
CASCADE_ORDER	STRING	"NONE", "FIRST", "LAST", "MIDDLE", "PARALLEL"	"NONE"	Specifies use, configuration and position of the cascade feature to bind more than one FIFO18E2 together to form deeper FIFO depths.
CLOCK_DOMAINS	STRING	"INDEPENDENT", "COMMON"	"INDEPENDENT"	Specifies whether to use a Common (synchronous operation) or Independent (asynchronous or different) clocks
FIRST_WORD_FALL_THROUGH	STRING	"FALSE", "TRUE"	"FALSE"	If TRUE, the first write to the FIFO will appear on DO without a first RDEN assertion.
INIT	HEX	Any 36-bit HEX value	All zeroes	Specifies the initial value on the DO output after configuration.
PROG_EMPTY_THRESH	DECIMAL	1 to 8191	256	Specifies the different between the write pointer (WRCOUNT) and read pointer (RDCOUNT). to trigger the PROGEMPTY output.
PROG_FULL_THRESH	DECIMAL	1 to 8191	256	Specifies the different between the write pointer (WRCOUNT) and read pointer (RDCOUNT). to trigger the PROGFULL output.
RDCOUNT_TYPE	STRING	"RAW_PNTR", "EXTENDED_DATACOUNT", "SIMPLE_DATACOUNT", "SYNC_PNTR"	"RAW_PNTR"	Specifies type and format for the RDCOUNT data.
READ_WIDTH	DECIMAL	4, 9, 18, 36	4	Specifies the data width for the read-side of the FIFO.
REGISTER_MODE	STRING	"UNREGISTERED", "DO_PIPELINED", "REGISTERED"	"UNREGISTERED"	Specifies output register mode.
RSTREG_PRIORITY	STRING	"RSTREG", "REGCE"	"RSTREG"	Specifies whether reset or enable has priority.
SLEEP_ASYNC	STRING	"FALSE", "TRUE"	"FALSE"	Specifies asynchronous or synchronous operation of the BRAM sleep mode.
SRVAL	HEX	Any 36-bit HEX value	All zeroes	Specifies per-bit the polarity of the FIFO output after RST/RSTREG is asserted
WRCOUNT_TYPE	STRING	"RAW_PNTR", "EXTENDED_DATACOUNT", "SIMPLE_DATACOUNT", "SYNC_PNTR"	"RAW_PNTR"	Specifies type and format for the WRCOUNT data.
WRITE_WIDTH	DECIMAL	4, 9, 18, 36	4	Specifies the data width for the write-side of the FIFO.

Attribute	Type	Allowed Values	Default	Description
<p><b>Programmable Inversion Attributes:</b> Specifies whether or not to use the optional inversions on specific pins of this component to change the active polarity of the pin function. When set to 1 on a clock pin (WRCLK or RDCLK), this components clocks on the negative edge. When set to 1 on other pins, it changes the function to behave active-Low rather than active-High.</p> <p>For pins that are buses, the bit-width of this attribute should match that of the bit-width of the associated pins and a binary value indicates which inverters to use and which to bypass.</p> <p>If an external inverter is specified on one of these associated pins, the Vivado Design Suite will automatically set this attribute during the opt_design stage so that additional logic is not necessary for changing the input polarity.</p>				
IS_RDCLK_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the RDCLK pin.
IS_RDEN_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the RDEN pin.
IS_RST_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the RST pin.
IS_RSTREG_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the RSTREG pin.
IS_WRCLK_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the WRCLK pin.
IS_WREN_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the WREN pin.

## VHDL Instantiation Template

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- FIFO18E2: 18Kb FIFO (First-In-First-Out) Block RAM Memory
-- UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

FIFO18E2_inst : FIFO18E2
generic map (
    CASCADE_ORDER => "NONE",           -- FIRST, LAST, MIDDLE, NONE, PARALLEL
    CLOCK_DOMAINS => "INDEPENDENT",    -- COMMON, INDEPENDENT
    FIRST_WORD_FALL_THROUGH => "FALSE", -- FALSE, TRUE
    INIT => X"0000000000",              -- Initial values on output port
    PROG_EMPTY_THRESH => 256,          -- Programmable Empty Threshold
    PROG_FULL_THRESH => 256,          -- Programmable Full Threshold
    -- Programmable Inversion Attributes: Specifies the use of the built-in programmable inversion
    IS_RDCLK_INVERTED => '0',          -- Optional inversion for RDCLK
    IS_RDEN_INVERTED => '0',          -- Optional inversion for RDEN
    IS_RSTREG_INVERTED => '0',        -- Optional inversion for RSTREG
    IS_RST_INVERTED => '0',           -- Optional inversion for RST
    IS_WRCLK_INVERTED => '0',         -- Optional inversion for WRCLK
    IS_WREN_INVERTED => '0',          -- Optional inversion for WREN
    RDCOUNT_TYPE => "RAW_PNTR",        -- EXTENDED_DATACOUNT, RAW_PNTR, SIMPLE_DATACOUNT, SYNC_PNTR
    READ_WIDTH => 4,                   -- 18-9

```

```

REGISTER_MODE => "UNREGISTERED",    -- DO_PIPELINED, REGISTERED, UNREGISTERED
RSTREG_PRIORITY => "RSTREG",        -- REGCE, RSTREG
SLEEP_ASYNC => "FALSE",            -- FALSE, TRUE
SRVAL => X"00000000",              -- SET/reset value of the FIFO outputs
WRCOUNT_TYPE => "RAW_PNTR",        -- EXTENDED_DATACOUNT, RAW_PNTR, SIMPLE_DATACOUNT, SYNC_PNTR
WRITE_WIDTH => 4                    -- 18-9
)
port map (
    -- Cascade Signals outputs: Multi-FIFO cascade signals
    CASDOUT => CASDOUT,              -- 32-bit output: Data cascade output bus
    CASDOUTP => CASDOUTP,            -- 4-bit output: Parity data cascade output bus
    CASNXTEMPTY => CASNXTEMPTY,      -- 1-bit output: Cascade next empty
    CASPRVRDEN => CASPRVRDEN,        -- 1-bit output: Cascade previous read enable
    -- Read Data outputs: Read output data
    DOUT => DOUT,                    -- 32-bit output: FIFO data output bus
    DOUTP => DOUTP,                  -- 4-bit output: FIFO parity output bus.
    -- Status outputs: Flags and other FIFO status outputs
    EMPTY => EMPTY,                  -- 1-bit output: Empty
    FULL => FULL,                    -- 1-bit output: Full
    PROGEMPTY => PROGEMPTY,          -- 1-bit output: Programmable empty
    PROGFULL => PROGFULL,            -- 1-bit output: Programmable full
    RDCOUNT => RDCOUNT,              -- 13-bit output: Read count
    RDERR => RDERR,                  -- 1-bit output: Read error
    RDRSTBUSY => RDRSTBUSY,          -- 1-bit output: Reset busy (sync to RDCLK)
    WRCOUNT => WRCOUNT,              -- 13-bit output: Write count
    WRERR => WRERR,                  -- 1-bit output: Write Error
    WRRSTBUSY => WRRSTBUSY,          -- 1-bit output: Reset busy (sync to WRCLK)
    -- Cascade Signals inputs: Multi-FIFO cascade signals
    CASDIN => CASDIN,                -- 32-bit input: Data cascade input bus
    CASDINP => CASDINP,              -- 4-bit input: Parity data cascade input bus
    CASDOMUX => CASDOMUX,            -- 1-bit input: Cascade MUX select
    CASDOMUXEN => CASDOMUXEN,        -- 1-bit input: Enable for cascade MUX select
    CASNXRDEN => CASNXRDEN,          -- 1-bit input: Cascade next read enable
    CASOREGIMUX => CASOREGIMUX,      -- 1-bit input: Cascade output MUX select
    CASOREGIMUXEN => CASOREGIMUXEN, -- 1-bit input: Cascade output MUX select enable
    CASPRVEMPTY => CASPRVEMPTY,      -- 1-bit input: Cascade previous empty
    -- Read Control Signals inputs: Read clock, enable and reset input signals
    RDCLK => RDCLK,                  -- 1-bit input: Read clock
    RDEN => RDEN,                    -- 1-bit input: Read enable
    REGCE => REGCE,                  -- 1-bit input: Output register clock enable
    RSTREG => RSTREG,                -- 1-bit input: Output register reset
    SLEEP => SLEEP,                  -- 1-bit input: Sleep Mode
    -- Write Control Signals inputs: Write clock and enable input signals
    RST => RST,                       -- 1-bit input: Reset
    WRCLK => WRCLK,                  -- 1-bit input: Write clock
    WREN => WREN,                    -- 1-bit input: Write enable
    -- Write Data inputs: Write input data
    DIN => DIN,                       -- 32-bit input: FIFO data input bus
    DINP => DINP,                     -- 4-bit input: FIFO parity input bus
);
-- End of FIFO18E2_inst instantiation

```

## Verilog Instantiation Template

```

// FIFO18E2: 18Kb FIFO (First-In-First-Out) Block RAM Memory
//      UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

FIFO18E2 #(
    .CASCADE_ORDER("NONE"),          // FIRST, LAST, MIDDLE, NONE, PARALLEL
    .CLOCK_DOMAINS("INDEPENDENT"),  // COMMON, INDEPENDENT
    .FIRST_WORD_FALL_THROUGH("FALSE"), // FALSE, TRUE
    .INIT(36'h00000000),             // Initial values on output port
    .PROG_EMPTY_THRESH(256),         // Programmable Empty Threshold
    .PROG_FULL_THRESH(256),          // Programmable Full Threshold
    // Programmable Inversion Attributes: Specifies the use of the built-in programmable inversion
    .IS_RDCLK_INVERTED(1'b0),        // Optional inversion for RDCLK
    .IS_RDEN_INVERTED(1'b0),         // Optional inversion for RDEN
    .IS_RSTREG_INVERTED(1'b0),       // Optional inversion for RSTREG

```



```

        .IS_RST_INVERTED(1'b0),           // Optional inversion for RST
        .IS_WRCLK_INVERTED(1'b0),       // Optional inversion for WRCLK
        .IS_WREN_INVERTED(1'b0),       // Optional inversion for WREN
        .RDCOUNT_TYPE("RAW_PNTR"),      // EXTENDED_DATACOUNT, RAW_PNTR, SIMPLE_DATACOUNT, SYNC_PNTR
        .READ_WIDTH(4),                 // 18-9
        .REGISTER_MODE("UNREGISTERED"), // DO_PIPELINED, REGISTERED, UNREGISTERED
        .RSTREG_PRIORITY("RSTREG"),     // REGCE, RSTREG
        .SLEEP_ASYNC("FALSE"),         // FALSE, TRUE
        .SRVAL(36'h00000000),          // SET/reset value of the FIFO outputs
        .WRCOUNT_TYPE("RAW_PNTR"),      // EXTENDED_DATACOUNT, RAW_PNTR, SIMPLE_DATACOUNT, SYNC_PNTR
        .WRITE_WIDTH(4)                 // 18-9
    )
    FIFO18E2_inst (
        // Cascade Signals outputs: Multi-FIFO cascade signals
        .CASDOUT(CASDOUT),               // 32-bit output: Data cascade output bus
        .CASDOUTP(CASDOUTP),            // 4-bit output: Parity data cascade output bus
        .CASNXEMPTY(CASNXEMPTY),        // 1-bit output: Cascade next empty
        .CASPRVRDEN(CASPRVRDEN),        // 1-bit output: Cascade previous read enable
        // Read Data outputs: Read output data
        .DOUT(DOUT),                    // 32-bit output: FIFO data output bus
        .DOUTP(DOUTP),                  // 4-bit output: FIFO parity output bus.
        // Status outputs: Flags and other FIFO status outputs
        .EMPTY(EMPTY),                  // 1-bit output: Empty
        .FULL(FULL),                    // 1-bit output: Full
        .PROGEMPTY(PROGEMPTY),          // 1-bit output: Programmable empty
        .PROGFULL(PROGFULL),            // 1-bit output: Programmable full
        .RDCOUNT(RDCOUNT),              // 13-bit output: Read count
        .RDERR(RDERR),                  // 1-bit output: Read error
        .RDRSTBUSY(RDRSTBUSY),          // 1-bit output: Reset busy (sync to RDCLK)
        .WRCOUNT(WRCOUNT),              // 13-bit output: Write count
        .WRERR(WRERR),                  // 1-bit output: Write Error
        .WRRSTBUSY(WRRSTBUSY),          // 1-bit output: Reset busy (sync to WRCLK)
        // Cascade Signals inputs: Multi-FIFO cascade signals
        .CASDIN(CASDIN),                // 32-bit input: Data cascade input bus
        .CASDINP(CASDINP),              // 4-bit input: Parity data cascade input bus
        .CASDOMUX(CASDOMUX),            // 1-bit input: Cascade MUX select
        .CASDOMUXEN(CASDOMUXEN),        // 1-bit input: Enable for cascade MUX select
        .CASNXTRDEN(CASNXTRDEN),        // 1-bit input: Cascade next read enable
        .CASOREGIMUX(CASOREGIMUX),      // 1-bit input: Cascade output MUX select
        .CASOREGIMUXEN(CASOREGIMUXEN), // 1-bit input: Cascade output MUX select enable
        .CASPRVEMPTY(CASPRVEMPTY),      // 1-bit input: Cascade previous empty
        // Read Control Signals inputs: Read clock, enable and reset input signals
        .RDCLK(RDCLK),                  // 1-bit input: Read clock
        .RDEN(RDEN),                    // 1-bit input: Read enable
        .REGCE(REGCE),                  // 1-bit input: Output register clock enable
        .RSTREG(RSTREG),                // 1-bit input: Output register reset
        .SLEEP(SLEEP),                  // 1-bit input: Sleep Mode
        // Write Control Signals inputs: Write clock and enable input signals
        .RST(RST),                       // 1-bit input: Reset
        .WRCLK(WRCLK),                   // 1-bit input: Write clock
        .WREN(WREN),                     // 1-bit input: Write enable
        // Write Data inputs: Write input data
        .DIN(DIN),                       // 32-bit input: FIFO data input bus
        .DINP(DINP)                       // 4-bit input: FIFO parity input bus
    );

    // End of FIFO18E2_inst instantiation

```

## For More Information

- See the *UltraScale Architecture Memory Resources User Guide* ([UG573](#)).
- See the [UltraScale Programmable Devices Documentation](#).

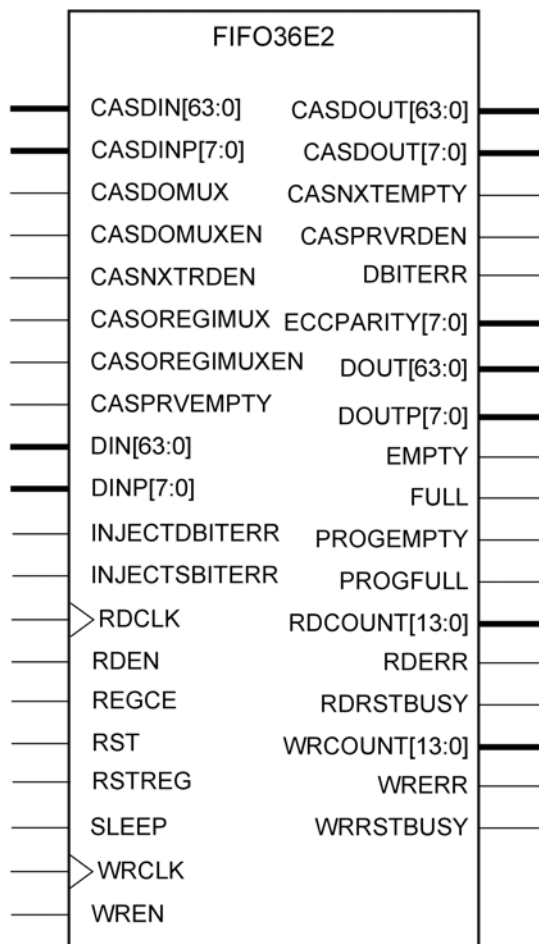
# FIFO36E2

Primitive: 36Kb FIFO (First-In-First-Out) Block RAM Memory

PRIMITIVE\_GROUP: BLOCKRAM

PRIMITIVE\_SUBGROUP: FIFO

Families: UltraScale, UltraScale+



X13403

## Introduction

The FIFO36E2 uses dedicated control logic and the 36Kb Block RAM to deliver a configurable First-In-First-Out (FIFO) capability. This primitive can be used in a 4-bit wide by 8K deep, 9-bit wide by 4K deep, 18-bit wide by 2K deep, 36-bit by 1K deep or a 72-bit wide by 512 deep configuration. The primitive can be configured in synchronous or dual-clock (asynchronous) mode, with all associated FIFO flags and status signals. This FIFO also features a cascade capability which lets you chain multiple FIFO36E2 components to form deeper FIFO configurations if desired.

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.

## Port Descriptions

Port	Direction	Width	Function
<b>Cascade Signals:</b> Signals used when cascading more than one FIFO			
CASDIN<63:0>	Input	64	Data input bus from previous CASDOUT bus when cascading FIFOs serially or in parallel to extend depth. Only used when CASCADE_ORDER="MIDDLE", "LAST", or "PARALLEL".
CASDINP<7:0>	Input	8	Parity data input bus from previous CASDOUTP bus when cascading FIFOs serially or in parallel to extend depth. Only used when CASCADE_ORDER="MIDDLE", "LAST", or "PARALLEL".
CASDOMUX	Input	1	D input to flop that drives the select line to the cascade mux on the Block RAM outputs. Only used when CASCADE_ORDER="PARALLEL".
CASDOMUXEN	Input	1	EN input to flop that drives the select line to the cascade mux on the Block RAM outputs. Only used when CASCADE_ORDER="PARALLEL".
CASDOUT<63:0>	Output	64	Data output bus to next CASDIN bus when cascading FIFOs serially or in parallel to extend depth. Only used when CASCADE_ORDER="FIRST", "MIDDLE", or "PARALLEL".
CASDOUTP<7:0>	Output	8	Parity data output bus to next CASDINP bus when cascading FIFOs serially or in parallel to extend depth. Only used when CASCADE_ORDER="FIRST", "MIDDLE", or "PARALLEL".
CASNXTEMPTY	Output	1	EMPTY input from previous FIFO, used for cascading FIFOs serially to extend depth. Connects to the CASNXTEMPTY of the previous FIFO. Only used when CASCADE_ORDER="MIDDLE" or "LAST".
CASNXRDEN	Input	1	RDEN input from next FIFO, used for cascading FIFOs serially to extend depth. Connects to CASPRVDEN of the next FIFO. Only used when CASCADE_ORDER="FIRST" or "MIDDLE".
CASOREGIMUX	Input	1	D input to flop that drives the select line to the cascade mux before the output registers. Only used when CASCADE_ORDER="PARALLEL" and REGISTER_MODE="DO_PIPELINED".
CASOREGIMUXEN	Input	1	EN input to flop that drives the select line to the cascade mux before the output registers. Only used when CASCADE_ORDER="PARALLEL" and REGISTER_MODE="DO_PIPELINED".
CASPRVEMPTY	Input	1	EMPTY input from previous FIFO, used for cascading FIFOs serially to extend depth. Connects to the CASNXTEMPTY of the previous FIFO. Only used when CASCADE_ORDER="MIDDLE" or "LAST".
CASPRVRDEN	Output	1	Control output driving the RDEN input of the previous (PRV) FIFO, used for cascading FIFOs serially to extend depth. Connects to CASNXTRDEN of the previous FIFO. Only used when CASCADE_ORDER="MIDDLE" or "LAST".
<b>ECC Signals:</b> Error Correction Circuitry ports			
DBITERR	Output	1	Status output from ECC function to indicate a double bit error was detected during a read operation. EN_ECC_READ needs to be TRUE in order to use this functionality. Synchronous to RDCLK.

Port	Direction	Width	Function
ECCPARITY<7:0>	Output	8	8-bit data generated by the ECC encoder used by the ECC decoder for memory error detection and correction. Applicable when EN_ECC_WRITE=1. Synchronous to WRCLK.
INJECTDBITERR	Input	1	Applicable when EN_ECC_WRITE=1. Causes a double-bit error to be inserted on bits 30 and 62 of DI during a write operation. Synchronous to WRCLK.
INJECTSBITERR	Input	1	Applicable when EN_ECC_WRITE=1. Causes a single-bit error to be inserted on bit 30 of DI during a write operation.
SBITERR	Output	1	ECC output indicating that a single-bit error was detected during the read operation. Synchronous to RDCLK.
<b>Read Control Signals:</b> Read clock, enable and reset input signals			
RDCLK	Input	1	Read clock.
RDEN	Input	1	Active-High read enable.
REGCE	Input	1	Active-High enable for output register stage.
RSTREG	Input	1	Active-High enable for output register reset.
SLEEP	Input	1	Dynamic shut down power saving. If SLEEP is High, the block is in power saving mode. If SLEEP_ASYNC=FALSE, synchronous to RDCLK, otherwise, asynchronous input.
<b>Read Data:</b> Read output data			
DOUT<63:0>	Output	64	FIFO data output bus. Synchronous to RDCLK.
DOUTP<7:0>	Output	8	FIFO parity output bus. Synchronous to RDCLK.
<b>Status:</b> Flags and other FIFO status outputs			
EMPTY	Output	1	Active-High flag to indicate when the FIFO is empty. Synchronous to RDCLK.
FULL	Output	1	Active-High flag to indicate when the FIFO is full. Synchronous to WRCLK.
PROGEMPTY	Output	1	Programmable flag to indicate the FIFO is almost empty (contains less than or equal to the number of words specified by the PROG_EMPTY_THRESH). Synchronous to RDCLK.
PROGFULL	Output	1	Programmable flag to indicate the FIFO is almost full (contains greater than or equal to the number of words specified by the PROG_FULL_THRESH) Synchronous to WRCLK.
RDCOUNT<13:0>	Output	14	Output of the internal FIFO read pointer, or a count of the number of words in the FIFO. Synchronous to RDCLK. Output value controlled by RDCOUNT_TYPE.
RDERR	Output	1	Indicates that a read operation failed due to FIFO being EMPTY, or FIFO being in a reset condition. Synchronous to RDCLK.
RDRSTBUSY	Output	1	Active-High indicator that the FIFO is currently in a reset state. Synchronous to RDCLK.
WRCOUNT<13:0>	Output	14	Output of the internal FIFO write pointer, or a count of the number of words in the FIFO. Synchronous to WRCLK. Output value controlled by WRCOUNT_TYPE.
WRERR	Output	1	Indicates that a write operation failed due to FIFO being FULL, or FIFO being in a reset condition. Synchronous to WRCLK.

Port	Direction	Width	Function
WRRSTBUSY	Output	1	Active-High indicator that the FIFO is currently in a reset state. Synchronous to WRCLK.
<b>Write Control Signals:</b> Write clock and enable input signals			
RST	Input	1	Active-High synchronous reset. Synchronous to WRCLK.
WRCLK	Input	1	Write clock.
WREN	Input	1	Active-High write enable.
<b>Write Data:</b> Write input data			
DIN<63:0>	Input	64	FIFO data input bus. Synchronous to WRCLK.
DINP<7:0>	Input	8	FIFO parity input bus. Synchronous to WRCLK.

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	Yes

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
CASCADE_ORDER	STRING	"NONE", "FIRST", "LAST", "MIDDLE", "PARALLEL"	"NONE"	Specifies use, configuration and position of the cascade feature to bind more than one FIFO36E2 together to form deeper FIFO depths.
CLOCK_DOMAINS	STRING	"INDEPENDENT", "COMMON"	"INDEPENDENT"	Specifies whether to use a Common (synchronous operation) or Independent (asynchronous or different) clocks
EN_ECC_PIPE	STRING	"FALSE", "TRUE"	"FALSE"	Enable ECC pipeline output register stage
EN_ECC_READ	STRING	"FALSE", "TRUE"	"FALSE"	Enable the ECC read decoder circuitry. Only valid when READ_WIDTH is set to 72.
EN_ECC_WRITE	STRING	"FALSE", "TRUE"	"FALSE"	Enable the ECC write encoder circuitry. Only valid when WRITE_WIDTH is set to 72.
FIRST_WORD_FALL_THROUGH	STRING	"FALSE", "TRUE"	"FALSE"	If TRUE, the first write to the FIFO will appear on DO without a first RDEN assertion.
INIT	HEX	Any 72-bit HEX value	All zeroes	Specifies the initial value on the DO output after configuration.

Attribute	Type	Allowed Values	Default	Description
PROG_EMPTY_THRESH	DECIMAL	1 to 8191	256	Specifies the different between the write pointer (WRCOUNT) and read pointer (RDCOUNT). to trigger the PROGEMPTY output.
PROG_FULL_THRESH	DECIMAL	1 to 8191	256	Specifies the different between the write pointer (WRCOUNT) and read pointer (RDCOUNT). to trigger the PROGFULL output.
RDCOUNT_TYPE	STRING	"RAW_PNTR", "EXTENDED_DATACOUNT", "SIMPLE_DATACOUNT", "SYNC_PNTR"	"RAW_PNTR"	Specifies type and format for the RDCOUNT data.
READ_WIDTH	DECIMAL	4, 9, 18, 36, 72	4	Specifies the data width for the read-side of the FIFO.
REGISTER_MODE	STRING	"UNREGISTERED", "DO_PIPELINED", "REGISTERED"	"UNREGISTERED"	Specifies output register mode.
RSTREG_PRIORITY	STRING	"RSTREG", "REGCE"	"RSTREG"	Specifies whether reset or enable has priority.
SLEEP_ASYNC	STRING	"FALSE", "TRUE"	"FALSE"	Specifies asynchronous or synchronous operation of the BRAM sleep mode.
SRVAL	HEX	Any 72-bit HEX value	All zeroes	Specifies per-bit the polarity of the FIFO output after RST/RSTREG is asserted
WRCOUNT_TYPE	STRING	"RAW_PNTR", "EXTENDED_DATACOUNT", "SIMPLE_DATACOUNT", "SYNC_PNTR"	"RAW_PNTR"	Specifies type and format for the WRCOUNT data.
WRITE_WIDTH	DECIMAL	4, 9, 18, 36, 72	4	Specifies the data width for the write-side of the FIFO.
<p><b>Programmable Inversion Attributes:</b> Specifies whether or not to use the optional inversions on specific pins of this component to change the active polarity of the pin function. When set to 1 on a clock pin (WRCLK or RDCLK), this components clocks on the negative edge. When set to 1 on other pins, it changes the function to behave active-Low rather than active-High. For pins that are buses, the bit-width of this attribute should match that of the bit-width of the associated pins and a binary value indicates which inverters to use and which to bypass. If an external inverter is specified on one of these associated pins, the Vivado Design Suite will automatically set this attribute during the opt_design stage so that additional logic is not necessary for changing the input polarity.</p>				
IS_RDCLK_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the RDCLK pin.
IS_RDEN_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the RDEN pin.

Attribute	Type	Allowed Values	Default	Description
IS_RST_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the RST pin.
IS_RSTREG_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the RSTREG pin.
IS_WRCLK_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the WRCLK pin.
IS_WREN_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the WREN pin.

## VHDL Instantiation Template

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- FIFO36E2: 36Kb FIFO (First-In-First-Out) Block RAM Memory
-- UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

FIFO36E2_inst : FIFO36E2
generic map (
    CASCADE_ORDER => "NONE",           -- FIRST, LAST, MIDDLE, NONE, PARALLEL
    CLOCK_DOMAINS => "INDEPENDENT",    -- COMMON, INDEPENDENT
    EN_ECC_PIPE => "FALSE",            -- ECC pipeline register, (FALSE, TRUE)
    EN_ECC_READ => "FALSE",           -- Enable ECC decoder, (FALSE, TRUE)
    EN_ECC_WRITE => "FALSE",          -- Enable ECC encoder, (FALSE, TRUE)
    FIRST_WORD_FALL_THROUGH => "FALSE", -- FALSE, TRUE
    INIT => X"00000000000000000000",    -- Initial values on output port
    PROG_EMPTY_THRESH => 256,         -- Programmable Empty Threshold
    PROG_FULL_THRESH => 256,         -- Programmable Full Threshold
    -- Programmable Inversion Attributes: Specifies the use of the built-in programmable inversion
    IS_RDCLK_INVERTED => '0',         -- Optional inversion for RDCLK
    IS_RDEN_INVERTED => '0',         -- Optional inversion for RDEN
    IS_RSTREG_INVERTED => '0',       -- Optional inversion for RSTREG
    IS_RST_INVERTED => '0',         -- Optional inversion for RST
    IS_WRCLK_INVERTED => '0',       -- Optional inversion for WRCLK
    IS_WREN_INVERTED => '0',       -- Optional inversion for WREN
    RDCOUNT_TYPE => "RAW_PNTR",      -- EXTENDED_DATACOUNT, RAW_PNTR, SIMPLE_DATACOUNT, SYNC_PNTR
    READ_WIDTH => 4,                 -- 18-9
    REGISTER_MODE => "UNREGISTERED", -- DO_PIPELINED, REGISTERED, UNREGISTERED
    RSTREG_PRIORITY => "RSTREG",     -- REGCE, RSTREG
    SLEEP_ASYNC => "FALSE",         -- FALSE, TRUE
    SRVAL => X"00000000000000000000", -- SET/reset value of the FIFO outputs
    WRCOUNT_TYPE => "RAW_PNTR",    -- EXTENDED_DATACOUNT, RAW_PNTR, SIMPLE_DATACOUNT, SYNC_PNTR
    WRITE_WIDTH => 4                 -- 18-9
)
port map (
    -- Cascade Signals outputs: Multi-FIFO cascade signals
    CASDOUT => CASDOUT,              -- 64-bit output: Data cascade output bus
    CASDOUTP => CASDOUTP,           -- 8-bit output: Parity data cascade output bus
    CASNXEMPTY => CASNXEMPTY,       -- 1-bit output: Cascade next empty
    CASPRVRDEN => CASPRVRDEN,       -- 1-bit output: Cascade previous read enable
    -- ECC Signals outputs: Error Correction Circuitry ports
    DBITERR => DBITERR,             -- 1-bit output: Double bit error status
    ECCPARITY => ECCPARITY,         -- 8-bit output: Generated error correction parity
    SBITERR => SBITERR,            -- 1-bit output: Single bit error status

```

```

-- Read Data outputs: Read output data
DOUT => DOUT,           -- 64-bit output: FIFO data output bus
DOUTP => DOUTP,         -- 8-bit output: FIFO parity output bus.
-- Status outputs: Flags and other FIFO status outputs
EMPTY => EMPTY,        -- 1-bit output: Empty
FULL => FULL,           -- 1-bit output: Full
PROGEMPTY => PROGEMPTY, -- 1-bit output: Programmable empty
PROGFULL => PROGFULL,   -- 1-bit output: Programmable full
RDCOUNT => RDCOUNT,     -- 14-bit output: Read count
RDERR => RDERR,         -- 1-bit output: Read error
RDRSTBUSY => RDRSTBUSY, -- 1-bit output: Reset busy (sync to RDCLK)
WRCOUNT => WRCOUNT,     -- 14-bit output: Write count
WRERR => WRERR,         -- 1-bit output: Write Error
WRRSTBUSY => WRRSTBUSY, -- 1-bit output: Reset busy (sync to WRCLK)
-- Cascade Signals inputs: Multi-FIFO cascade signals
CASDIN => CASDIN,       -- 64-bit input: Data cascade input bus
CASDINP => CASDINP,     -- 8-bit input: Parity data cascade input bus
CASDOMUX => CASDOMUX,   -- 1-bit input: Cascade MUX select input
CASDOMUXEN => CASDOMUXEN, -- 1-bit input: Enable for cascade MUX select
CASNXTRDEN => CASNXTRDEN, -- 1-bit input: Cascade next read enable
CASOREGIMUX => CASOREGIMUX, -- 1-bit input: Cascade output MUX select
CASOREGIMUXEN => CASOREGIMUXEN, -- 1-bit input: Cascade output MUX select enable
CASPRVEMPTY => CASPRVEMPTY, -- 1-bit input: Cascade previous empty
-- ECC Signals inputs: Error Correction Circuitry ports
INJECTDBITERR => INJECTDBITERR, -- 1-bit input: Inject a double bit error
INJECTSBITERR => INJECTSBITERR, -- 1-bit input: Inject a single bit error
-- Read Control Signals inputs: Read clock, enable and reset input signals
RDCLK => RDCLK,         -- 1-bit input: Read clock
RDEN => RDEN,           -- 1-bit input: Read enable
REGCE => REGCE,         -- 1-bit input: Output register clock enable
RSTREG => RSTREG,       -- 1-bit input: Output register reset
SLEEP => SLEEP,         -- 1-bit input: Sleep Mode
-- Write Control Signals inputs: Write clock and enable input signals
RST => RST,             -- 1-bit input: Reset
WRCLK => WRCLK,         -- 1-bit input: Write clock
WREN => WREN,           -- 1-bit input: Write enable
-- Write Data inputs: Write input data
DIN => DIN,             -- 64-bit input: FIFO data input bus
DINP => DINP,           -- 8-bit input: FIFO parity input bus
);

-- End of FIFO36E2_inst instantiation
    
```

## Verilog Instantiation Template

```

// FIFO36E2: 36Kb FIFO (First-In-First-Out) Block RAM Memory
// UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

FIFO36E2 #(
    .CASCADE_ORDER("NONE"),           // FIRST, LAST, MIDDLE, NONE, PARALLEL
    .CLOCK_DOMAINS("INDEPENDENT"),    // COMMON, INDEPENDENT
    .EN_ECC_PIPE("FALSE"),            // ECC pipeline register, (FALSE, TRUE)
    .EN_ECC_READ("FALSE"),            // Enable ECC decoder, (FALSE, TRUE)
    .EN_ECC_WRITE("FALSE"),           // Enable ECC encoder, (FALSE, TRUE)
    .FIRST_WORD_FALL_THROUGH("FALSE"), // FALSE, TRUE
    .INIT(72'h00000000000000000000),  // Initial values on output port
    .PROG_EMPTY_THRESH(256),          // Programmable Empty Threshold
    .PROG_FULL_THRESH(256),           // Programmable Full Threshold
    // Programmable Inversion Attributes: Specifies the use of the built-in programmable inversion
    .IS_RDCLK_INVERTED(1'b0),         // Optional inversion for RDCLK
    .IS_RDEN_INVERTED(1'b0),          // Optional inversion for RDEN
    .IS_RSTREG_INVERTED(1'b0),        // Optional inversion for RSTREG
    .IS_RST_INVERTED(1'b0),           // Optional inversion for RST
    .IS_WRCLK_INVERTED(1'b0),         // Optional inversion for WRCLK
    .IS_WREN_INVERTED(1'b0),          // Optional inversion for WREN
    .RDCOUNT_TYPE("RAW_PNTR"),         // EXTENDED_DATACOUNT, RAW_PNTR, SIMPLE_DATACOUNT, SYNC_PNTR
    .READ_WIDTH(4),                   // 18-9
    .REGISTER_MODE("UNREGISTERED"),   // DO_PIPELINED, REGISTERED, UNREGISTERED
    .RSTREG_PRIORITY("RSTREG"),        // REGCE, RSTREG
    )
    
```



```

        .SLEEP_ASYNC("FALSE"),           // FALSE, TRUE
        .SRVAL(72'h0000000000000000),    // SET/reset value of the FIFO outputs
        .WRCOUNT_TYPE("RAW_PNTR"),       // EXTENDED_DATACOUNT, RAW_PNTR, SIMPLE_DATACOUNT, SYNC_PNTR
        .WRITE_WIDTH(4)                  // 18-9
    )
    FIFO36E2_inst (
        // Cascade Signals outputs: Multi-FIFO cascade signals
        .CASDOUT(CASDOUT),                // 64-bit output: Data cascade output bus
        .CASDOUTP(CASDOUTP),              // 8-bit output: Parity data cascade output bus
        .CASNXTEMPY(CASNXTEMPY),          // 1-bit output: Cascade next empty
        .CASPRVRDEN(CASPRVRDEN),          // 1-bit output: Cascade previous read enable
        // ECC Signals outputs: Error Correction Circuitry ports
        .DBITERR(DBITERR),                // 1-bit output: Double bit error status
        .ECCPARITY(ECCPARITY),            // 8-bit output: Generated error correction parity
        .SBITERR(SBITERR),                // 1-bit output: Single bit error status
        // Read Data outputs: Read output data
        .DOUT(DOUT),                      // 64-bit output: FIFO data output bus
        .DOUTP(DOUTP),                    // 8-bit output: FIFO parity output bus.
        // Status outputs: Flags and other FIFO status outputs
        .EMPTY(EMPTY),                    // 1-bit output: Empty
        .FULL(FULL),                      // 1-bit output: Full
        .PROGEMPTY(PROGEMPTY),            // 1-bit output: Programmable empty
        .PROGFULL(PROGFULL),              // 1-bit output: Programmable full
        .RDCCOUNT(RDCCOUNT),              // 14-bit output: Read count
        .RDERR(RDERR),                    // 1-bit output: Read error
        .RDRSTBUSY(RDRSTBUSY),            // 1-bit output: Reset busy (sync to RDCLK)
        .WRCOUNT(WRCOUNT),                // 14-bit output: Write count
        .WRERR(WRERR),                    // 1-bit output: Write Error
        .WRRSTBUSY(WRRSTBUSY),            // 1-bit output: Reset busy (sync to WRCLK)
        // Cascade Signals inputs: Multi-FIFO cascade signals
        .CASDIN(CASDIN),                  // 64-bit input: Data cascade input bus
        .CASDINP(CASDINP),                // 8-bit input: Parity data cascade input bus
        .CASDOMUX(CASDOMUX),              // 1-bit input: Cascade MUX select input
        .CASDOMUXEN(CASDOMUXEN),          // 1-bit input: Enable for cascade MUX select
        .CASNXTRDEN(CASNXTRDEN),          // 1-bit input: Cascade next read enable
        .CASOREGIMUX(CASOREGIMUX),        // 1-bit input: Cascade output MUX select
        .CASOREGIMUXEN(CASOREGIMUXEN),    // 1-bit input: Cascade output MUX select enable
        .CASPRVEMPTY(CASPRVEMPTY),        // 1-bit input: Cascade previous empty
        // ECC Signals inputs: Error Correction Circuitry ports
        .INJECTDBITERR(INJECTDBITERR),    // 1-bit input: Inject a double bit error
        .INJECTSBITERR(INJECTSBITERR),    // 1-bit input: Inject a single bit error
        // Read Control Signals inputs: Read clock, enable and reset input signals
        .RDCLK(RDCLK),                    // 1-bit input: Read clock
        .RDEN(RDEN),                      // 1-bit input: Read enable
        .REGCE(REGCE),                    // 1-bit input: Output register clock enable
        .RSTREG(RSTREG),                  // 1-bit input: Output register reset
        .SLEEP(SLEEP),                    // 1-bit input: Sleep Mode
        // Write Control Signals inputs: Write clock and enable input signals
        .RST(RST),                        // 1-bit input: Reset
        .WRCLK(WRCLK),                    // 1-bit input: Write clock
        .WREN(WREN),                      // 1-bit input: Write enable
        // Write Data inputs: Write input data
        .DIN(DIN),                        // 64-bit input: FIFO data input bus
        .DINP(DINP),                      // 8-bit input: FIFO parity input bus
    );

    // End of FIFO36E2_inst instantiation

```

## For More Information

- See the *UltraScale Architecture Memory Resources User Guide* ([UG573](#)).
- See the [UltraScale Programmable Devices Documentation](#).

# FRAME\_ECCE3

Primitive: Configuration Frame Error Correction

PRIMITIVE\_GROUP: CONFIGURATION

PRIMITIVE\_SUBGROUP: ECC

Families: UltraScale

## Introduction

This design element enables the dedicated, built-in Error Correction Code (ECC) for the configuration memory of the device. This element contains outputs that allow monitoring of the status of the ECC circuitry and the status of the readback CRC circuitry. This element is not intended to be instantiated, used, or modified outside of the Xilinx generated SEM IP

## Design Entry Method

Instantiation	No
Inference	No
IP and IP Integrator Catalog	Recommended

## For More Information

- See the *UltraScale Architecture Soft Error Mitigation Controller v3.0 Product Guide* ([PG187](#)).
- See the [UltraScale Programmable Devices Documentation](#).

## GTHE3\_CHANNEL

Primitive: Gigabit Transceiver for UltraScale devices

PRIMITIVE\_GROUP: [ADVANCED](#)

PRIMITIVE\_SUBGROUP: GT

Families: UltraScale

### Introduction

GTHE3 is a gigabit transceiver component for UltraScale devices. This element is not intended to be instantiated, used, or modified outside of Xilinx generated IP. See the Transceivers User Guide for details on this component.

### Design Entry Method

Instantiation	No
Inference	No
IP and IP Integrator Catalog	Recommended

### For More Information

- See the *UltraScale Architecture GTH Transceivers User Guide* ([UG576](#))
- See the *UltraScale FPGAs Transceivers Wizard v1.6 Product Guide* ([PG182](#)).
- See the [UltraScale Programmable Devices Documentation](#).

# GTHE3\_COMMON

Primitive: Gigabit Transceiver for UltraScale devices

PRIMITIVE\_GROUP: **ADVANCED**

PRIMITIVE\_SUBGROUP: GT

Families: UltraScale

## Introduction

GTHE3 is a gigabit transceiver component for UltraScale devices. This element is not intended to be instantiated, used, or modified outside of Xilinx generated IP. See the Transceivers User Guide for details on this component.

## Design Entry Method

Instantiation	No
Inference	No
IP and IP Integrator Catalog	Recommended

## For More Information

- See the *UltraScale Architecture GTH Transceivers User Guide* ([UG576](#))
- See the *UltraScale FPGAs Transceivers Wizard v1.6 Product Guide* ([PG182](#)).
- See the [UltraScale Programmable Devices Documentation](#).

# GTHE4\_CHANNEL

Primitive: Gigabit Transceiver for UltraScale+ devices

PRIMITIVE\_GROUP: [ADVANCED](#)

PRIMITIVE\_SUBGROUP: GT

Families: UltraScale+

## Introduction

GTHE4 is a gigabit transceiver component for UltraScale+ devices. This element is not intended to be instantiated, used, or modified outside of Xilinx generated IP. See the Transceivers User Guide for details on this component.

## Design Entry Method

Instantiation	No
Inference	No
IP and IP Integrator Catalog	Recommended

## For More Information

- See the *UltraScale Architecture GTH Transceivers User Guide* ([UG576](#))
- See the *UltraScale FPGAs Transceivers Wizard v1.6 Product Guide* ([PG182](#)).
- See the [UltraScale Programmable Devices Documentation](#).

# GTHE4\_COMMON

Primitive: Gigabit Transceiver for UltraScale+ devices

PRIMITIVE\_GROUP: [ADVANCED](#)

PRIMITIVE\_SUBGROUP: [GT](#)

Families: [UltraScale+](#)

## Introduction

GTHE4 is a gigabit transceiver component for UltraScale+ devices. This element is not intended to be instantiated, used, or modified outside of Xilinx generated IP. See the Transceivers User Guide for details on this component.

## Design Entry Method

Instantiation	No
Inference	No
IP and IP Integrator Catalog	Recommended

## For More Information

- See the *UltraScale Architecture GTH Transceivers User Guide* ([UG576](#))
- See the *UltraScale FPGAs Transceivers Wizard v1.6 Product Guide* ([PG182](#)).
- See the [UltraScale Programmable Devices Documentation](#).

# GTYE3\_CHANNEL

Primitive: Gigabit Transceiver for UltraScale devices

PRIMITIVE\_GROUP: [ADVANCED](#)

PRIMITIVE\_SUBGROUP: GT

Families: UltraScale

## Introduction

GTYE3 is a gigabit transceiver component for UltraScale devices. This element is not intended to be instantiated, used, or modified outside of Xilinx generated IP. See the Transceivers User Guide for details on this component.

## Design Entry Method

Instantiation	No
Inference	No
IP and IP Integrator Catalog	Recommended

## For More Information

- See the *UltraScale Architecture GTY Transceivers Advance Specification User Guide* ([UG578](#))
- See the *UltraScale FPGAs Transceivers Wizard v1.6 Product Guide* ([PG182](#)).
- See the [UltraScale Programmable Devices Documentation](#).

# GTYE3\_COMMON

Primitive: Gigabit Transceiver for UltraScale devices

PRIMITIVE\_GROUP: [ADVANCED](#)

PRIMITIVE\_SUBGROUP: GT

Families: UltraScale

## Introduction

GTYE3 is a gigabit transceiver component for UltraScale devices. This element is not intended to be instantiated, used, or modified outside of Xilinx generated IP. See the Transceivers User Guide for details on this component.

## Design Entry Method

Instantiation	No
Inference	No
IP and IP Integrator Catalog	Recommended

## For More Information

- See the *UltraScale Architecture GTY Transceivers Advance Specification User Guide* ([UG578](#))
- See the *UltraScale FPGAs Transceivers Wizard v1.6 Product Guide* ([PG182](#)).
- See the [UltraScale Programmable Devices Documentation](#).



## GTYE4\_CHANNEL

Primitive: Gigabit Transceiver for UltraScale+ devices

PRIMITIVE\_GROUP: [ADVANCED](#)

PRIMITIVE\_SUBGROUP: GT

Families: UltraScale+

### Introduction

GTYE4 is a gigabit transceiver component for UltraScale+ devices. This element is not intended to be instantiated, used, or modified outside of Xilinx generated IP. See the Transceivers User Guide for details on this component.

### Design Entry Method

Instantiation	No
Inference	No
IP and IP Integrator Catalog	Recommended

### For More Information

- See the *UltraScale Architecture GTY Transceivers Advance Specification User Guide* ([UG578](#))
- See the *UltraScale FPGAs Transceivers Wizard v1.6 Product Guide* ([PG182](#)).
- See the [UltraScale Programmable Devices Documentation](#).

# GTYE4\_COMMON

Primitive: Gigabit Transceiver for UltraScale+ devices

PRIMITIVE\_GROUP: [ADVANCED](#)

PRIMITIVE\_SUBGROUP: GT

Families: UltraScale+

## Introduction

GTYE4 is a gigabit transceiver component for UltraScale+ devices. This element is not intended to be instantiated, used, or modified outside of Xilinx generated IP. See the Transceivers User Guide for details on this component.

## Design Entry Method

Instantiation	No
Inference	No
IP and IP Integrator Catalog	Recommended

## For More Information

- See the *UltraScale Architecture GTY Transceivers Advance Specification User Guide* ([UG578](#))
- See the *UltraScale FPGAs Transceivers Wizard v1.6 Product Guide* ([PG182](#)).
- See the [UltraScale Programmable Devices Documentation](#).

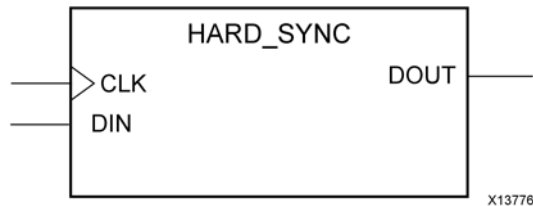
## HARD\_SYNC

Primitive: Metastability Hardened Registers

PRIMITIVE\_GROUP: REGISTER

PRIMITIVE\_SUBGROUP: METASTABILITY

Families: UltraScale, UltraScale+



### Introduction

Metastability hardened registers are generally used for asynchronous domain crossings to synchronize signals in which may incur setup or hold time violations. The LATENCY attribute may be set to configure a dual or triple register synchronizer configuration.

### Port Descriptions

Port	Direction	Width	Function
CLK	Input	1	Clock input
DIN	Input	1	Data input
DOUT	Output	1	Data output

### Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	No

### Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	BINARY	1'b0, 1'b1	1'b0	Specifies the initial values of the HARD_SYNC output upon completion of configuration and release of GSR.
IS_CLK_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies to use the programmable inversion on the CLK input to change the clock from active-High to active-Low.
LATENCY	DECIMAL	2, 3	2	Specifies whether to use a two stage or three stage synchronizer. Three stage will exhibit better MTBF characteristics however have an additional clock cycle of latency.

## VHDL Instantiation Template

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- HARD_SYNC: Metastability Hardened Registers
--           UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

HARD_SYNC_inst : HARD_SYNC
generic map (
    INIT => '0',           -- Initial values, '0', '1'
    IS_CLK_INVERTED => '0', -- Programmable inversion on CLK input
    LATENCY => 2           -- 2-3
)
port map (
    DOUT => DOUT, -- 1-bit output: Data
    CLK => CLK,   -- 1-bit input: Clock
    DIN => DIN    -- 1-bit input: Data
);

-- End of HARD_SYNC_inst instantiation
    
```

## Verilog Instantiation Template

```

// HARD_SYNC: Metastability Hardened Registers
//           UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

HARD_SYNC #(
    .INIT(1'b0),           // Initial values, 1'b0, 1'b1
    .IS_CLK_INVERTED(1'b0), // Programmable inversion on CLK input
    .LATENCY(2)           // 2-3
)
HARD_SYNC_inst (
    .DOUT(DOUT), // 1-bit output: Data
    .CLK(CLK),   // 1-bit input: Clock
    .DIN(DIN)    // 1-bit input: Data
);

// End of HARD_SYNC_inst instantiation
    
```

## For More Information

- See the *UltraScale Architecture Configurable Logic Block User Guide* ([UG574](#)).
- See the [UltraScale Programmable Devices Documentation](#).

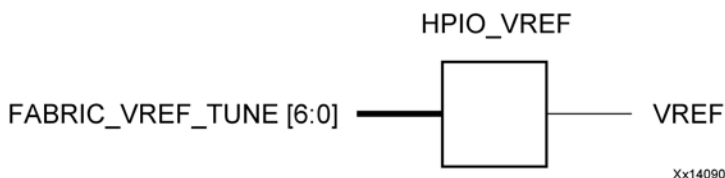
## HPIO\_VREF

Primitive: VREF Scan

PRIMITIVE\_GROUP: I/O

PRIMITIVE\_SUBGROUP: INPUT\_BUFFER

Families: UltraScale, UltraScale+



### Introduction

The VREF\_HPIO component used with the IBUFE3 or IOBUFE3 buffers provides access to the VREF scan capability in the HPIO banks.

### Port Descriptions

Port	Direction	Width	Function
FABRIC_VREF_TUNE<6:0>	Input	7	VREF tuning input value to allow Vref adjustment.
VREF	Output	1	Tuned output that connects to all associated IBUFE3 or IOBUFE3 components within an HPIO bank.

### Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	No

### Available Attributes

Attribute	Type	Allowed Values	Default	Description
VREF_CNTR	STRING	"OFF", "FABRIC_RANGE1", "FABRIC_RANGE2"	"OFF"	Specifies VREF counter range.

## VHDL Instantiation Template

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- HPIO_VREF: VREF Scan
--           UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

HPIO_VREF_inst : HPIO_VREF
generic map (
    VREF_CNTR => "OFF" -- FABRIC_RANGE1, FABRIC_RANGE2, OFF
)
port map (
    VREF => VREF, -- 1-bit output: Tuned output (connect to associated IBUFE3
                -- component)

    FABRIC_VREF_TUNE => FABRIC_VREF_TUNE -- 7-bit input: VREF tuning value
);

-- End of HPIO_VREF_inst instantiation
    
```

## Verilog Instantiation Template

```

// HPIO_VREF: VREF Scan
//           UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

HPIO_VREF #(
    .VREF_CNTR("OFF") // FABRIC_RANGE1, FABRIC_RANGE2, OFF
)
HPIO_VREF_inst (
    .VREF(VREF), // 1-bit output: Tuned output (connect to associated IBUFE3
                // component)

    .FABRIC_VREF_TUNE(FABRIC_VREF_TUNE) // 7-bit input: VREF tuning value
);

// End of HPIO_VREF_inst instantiation
    
```

## For More Information

- See the *UltraScale Architecture SelectIO Resources User Guide* ([UG571](#)).
- See the [UltraScale Programmable Devices Documentation](#).

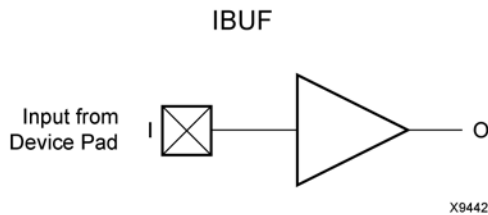
## IBUF

Primitive: Input Buffer

PRIMITIVE\_GROUP: I/O

PRIMITIVE\_SUBGROUP: INPUT\_BUFFER

Families: UltraScale, UltraScale+



### Introduction

Single-ended signals used as simple inputs must use an input buffer (IBUF).

I/O attributes that do not impact the logic function of the component, such as IOSTANDARD and IBUF\_LOW\_PWR, should be supplied to the top-level port via an appropriate property. For details on applying such properties to the associated port, see UG912: Vivado Design Suite Properties Reference Guide.

### Port Descriptions

Port	Direction	Width	Function
I	Input	1	Buffer Input connected to a top-level input port.
O	Output	1	Buffer output connected to internal device circuitry.

### Design Entry Method

Instantiation	Yes
Inference	Recommended
IP and IP Integrator Catalog	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;

-- IBUF: Input Buffer
--      UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

IBUF_inst : IBUF
port map (
    
```

```
O => O, -- 1-bit output: Buffer output
I => I -- 1-bit input: Buffer input
);

-- End of IBUF_inst instantiation
```

## Verilog Instantiation Template

```
// IBUF: Input Buffer
// UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

IBUF IBUF_inst (
    .O(O), // 1-bit output: Buffer output
    .I(I) // 1-bit input: Buffer input
);

// End of IBUF_inst instantiation
```

## For More Information

- See the *UltraScale Architecture SelectIO Resources User Guide* ([UG571](#)).
- See the [UltraScale Programmable Devices Documentation](#).



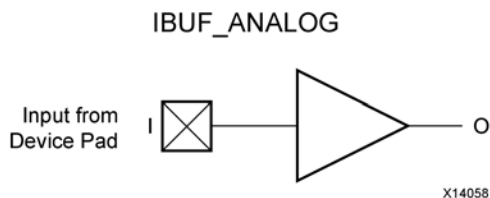
## IBUF\_ANALOG

Primitive: Analog Auxiliary SYSMON Input Buffer

PRIMITIVE\_GROUP: I/O

PRIMITIVE\_SUBGROUP: INPUT\_BUFFER

Families: UltraScale, UltraScale+



### Introduction

This design element is an input buffer used to connect the auxiliary analog inputs to the SYSMONE1 component. When using the VAUXP/VAUXN pins of the SYSMONE1 component, this buffer allows for a proper connection to the top-level port in the design.

### Port Descriptions

Port	Direction	Width	Function
I	Input	1	Connect this pin to a top-level port in the design.
O	Output	1	Connect this pin to the VAUXP or VAUXN port of a SYSMONE1 component.

### Design Entry Method

Instantiation	Yes
Inference	Yes
IP and IP Integrator Catalog	Recommended

### VHDL 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_ANALOG: Analog Auxiliary SYSMON Input Buffer
-- UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

IBUF_ANALOG_inst : IBUF_ANALOG
port map (
    O => O, -- 1-bit output: Connect to a VAUXP/VAUXN port of the SYSMONE1
    I => I -- 1-bit input: Connect to a top-level design port
);
    
```

```
-- End of IBUF_ANALOG_inst instantiation
```

## Verilog Instantiation Template

```
// IBUF_ANALOG: Analog Auxiliary SYSMON Input Buffer
//                               UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

IBUF_ANALOG IBUF_ANALOG_inst (
    .O(O), // 1-bit output: Connect to a VAUXP/VAUXN port of the SYSMONE1
    .I(I) // 1-bit input: Connect to a top-level design port
);

// End of IBUF_ANALOG_inst instantiation
```

## For More Information

- See the *UltraScale Architecture System Monitor User Guide* ([UG580](#)).
- See the [UltraScale Programmable Devices Documentation](#).

# IBUF\_IBUFDISABLE

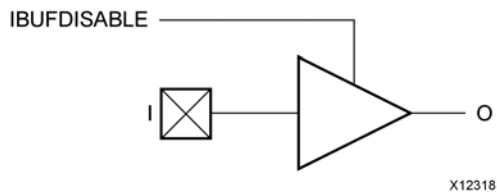
Primitive: Input Buffer With Input Buffer Disable

PRIMITIVE\_GROUP: I/O

PRIMITIVE\_SUBGROUP: INPUT\_BUFFER

Families: UltraScale, UltraScale+

IBUF\_IBUFDISABLE



## Introduction

The IBUF\_IBUFDISABLE primitive is an input buffer with a disable port that can be used as an additional power saving feature for periods when the input is not used.

The IBUF\_IBUFDISABLE primitive can disable the input buffer and force the O output to the internal logic to a logic-Low when the IBUFDISABLE signal is asserted High. The USE\_IBUFDISABLE attribute must be set to TRUE and SIM\_DEVICE to ULTRASCALE for this primitive to have the expected behavior specific to the UltraScale architecture. This feature can be used to reduce power at times when the I/O is idle. Input buffers that use the VREF power rail (such as SSTL and HSTL) benefit the most from the IBUFDISABLE being set to TRUE because they tend to have higher static power consumption than the non-VREF standards such as LVCMOS and LVTTL.

I/O attributes that do not impact the logic function of the component such as IOSTANDARD, DIFF\_TERM and IBUF\_LOW\_PWR should be supplied to the top-level port via an appropriate property. For details on applying such properties to the associated port, see UG912: Vivado Design Suite Properties Reference Guide.

## Port Descriptions

Port	Direction	Width	Function
I	Input	1	Input port connection. Connect directly to top-level port in the design.
IBUFDISABLE	Input	1	Disables input path through the buffer and forces to a logic low. This feature is generally used to reduce power at times when the I/O is idle for a period of time.
O	Output	1	Buffer output representing the input path to the device.

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
SIM_DEVICE	STRING	"ULTRASCALE", "7SERIES"	"7SERIES"	This attribute must be set to "ULTRASCALE" in order to exhibit the proper simulation behavior.
USE_IBUFDISABLE	STRING	"TRUE", "FALSE"	"TRUE"	This attribute must be unspecified or set to "TRUE" if specified.

## VHDL 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_IBUFDISABLE: Input Buffer With Input Buffer Disable
--                               UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

IBUF_IBUFDISABLE_inst : IBUF_IBUFDISABLE
generic map (
    SIM_DEVICE => "ULTRASCALE" -- Must be set to "ULTRASCALE"
)
port map (
    O => O,                -- 1-bit output: Buffer output
    I => I,                -- 1-bit input: Buffer input (connect directly to top-level port)
    IBUFDISABLE => IBUFDISABLE -- 1-bit input: Buffer disable input, high=disable
);

-- End of IBUF_IBUFDISABLE_inst instantiation
    
```

## Verilog Instantiation Template

```

// IBUF_IBUFDISABLE: Input Buffer With Input Buffer Disable
//                               UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

IBUF_IBUFDISABLE #(
    .SIM_DEVICE("ULTRASCALE") // Must be set to "ULTRASCALE"
)
IBUF_IBUFDISABLE_inst (
    .O(O),                // 1-bit output: Buffer output
    .I(I),                // 1-bit input: Buffer input (connect directly to top-level port)
    .IBUFDISABLE(IBUFDISABLE) // 1-bit input: Buffer disable input, high=disable
);

// End of IBUF_IBUFDISABLE_inst instantiation
    
```

## For More Information

- See the *UltraScale Architecture SelectIO Resources User Guide* ([UG571](#)).
- See the [UltraScale Programmable Devices Documentation](#).

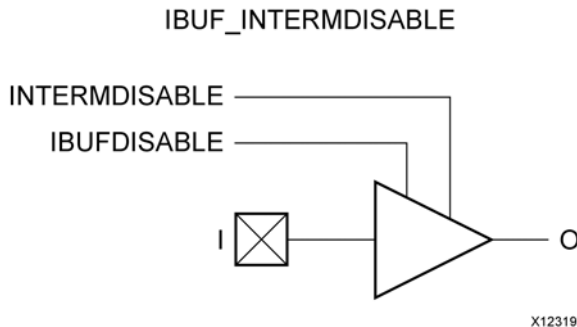
## IBUF\_INTERMDISABLE

**Primitive: Input Buffer With Input Buffer Disable and On-die Input Termination Disable**

PRIMITIVE\_GROUP: I/O

PRIMITIVE\_SUBGROUP: INPUT\_BUFFER

Families: UltraScale, UltraScale+



### Introduction

The IBUF\_INTERMDISABLE primitive is available in the HR I/O banks and is similar to the IBUF\_IBUFDISABLE primitive in that it has a IBUFDISABLE port that can be used to disable the input buffer during periods that the buffer is not being used. The USE\_IBUFDISABLE attribute must be set to TRUE and SIM\_DEVICE to ULTRASCALE for this primitive to have the expected behavior specific to the UltraScale architecture. The IBUF\_INTERMDISABLE primitive also has an INTERMDISABLE port that can be used to disable the optional on-die receiver termination feature. See UG571: UltraScale Architecture SelectIO Resources, "Uncalibrated Input Termination in I/O Banks" for more details.

The IBUF\_INTERMDISABLE primitive can disable the input buffer and force the O output to the internal logic to a logic-Low when the IBUFDISABLE signal is asserted High. The IBUF\_INTERMDISABLE primitive further allows the termination legs to be disabled whenever the INTERMDISABLE signal is asserted High. These features can be combined to reduce power whenever the input is idle. Input buffers that use the VREF power rail (such as SSTL and HSTL) benefit the most from the IBUFDISABLE signal being set to TRUE because they tend to have higher static power consumption than the non-VREF standards such as LVCMOS and LVTTTL.

I/O attributes that do not impact the logic function of the component, such as IOSTANDARD and IBUF\_LOW\_PWR, should be supplied to the top-level port via an appropriate property. For details on applying such properties to the associated port, see UG912: Vivado Design Suite Properties Reference Guide.

## Port Descriptions

Port	Direction	Width	Function
I	Input	1	Input port connection. Connect directly to top-level port in the design.
IBUFDISABLE	Input	1	Disables input path through the buffer and forces to a logic low. This feature is generally used to reduce power at times when the I/O is idle for a period of time.
INTERMDISABLE	Input	1	Control to enable/disable of on-chip input termination. This is generally used to reduce power in long periods of an idle state.
O	Output	1	Buffer output representing the input path to the device.

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
SIM_DEVICE	STRING	"ULTRASCALE", "7SERIES"	"7SERIES"	This attribute must be set to "ULTRASCALE" in order to exhibit the proper simulation behavior.
USE_IBUFDISABLE	STRING	"TRUE", "FALSE"	"TRUE"	This attribute must be unspecified or set to "TRUE" if specified.

## VHDL 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_INTERMDISABLE: Input Buffer With Input Buffer Disable and On-die Input Termination Disable
--                               UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

IBUF_INTERMDISABLE_inst : IBUF_INTERMDISABLE
generic map (
    SIM_DEVICE => "ULTRASCALE" -- Must be set to "ULTRASCALE"
)
port map (
    O => O,                -- 1-bit output: Buffer output
    I => I,                -- 1-bit input: Buffer input (connect directly to top-level port)
    IBUFDISABLE => IBUFDISABLE, -- 1-bit input: Buffer disable input, high=disable
    INTERMDISABLE => INTERMDISABLE -- 1-bit input: Input Termination Disable
);

-- End of IBUF_INTERMDISABLE_inst instantiation
    
```

## Verilog Instantiation Template

```
// IBUF_INTERMDISABLE: Input Buffer With Input Buffer Disable and On-die Input Termination Disable
//                               UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

IBUF_INTERMDISABLE #(
    .SIM_DEVICE("ULTRASCALE") // Must be set to "ULTRASCALE"
)
IBUF_INTERMDISABLE_inst (
    .O(O), // 1-bit output: Buffer output
    .I(I), // 1-bit input: Buffer input (connect directly to top-level port)
    .IBUFDISABLE(IBUFDISABLE), // 1-bit input: Buffer disable input, high=disable
    .INTERMDISABLE(INTERMDISABLE) // 1-bit input: Input Termination Disable
);

// End of IBUF_INTERMDISABLE_inst instantiation
```

## For More Information

- See the *UltraScale Architecture SelectIO Resources User Guide* ([UG571](#)).
- See the [UltraScale Programmable Devices Documentation](#).



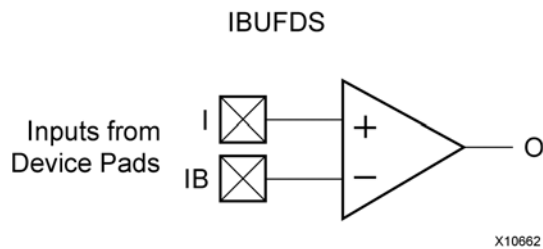
## IBUFDS

Primitive: Differential Input Buffer

PRIMITIVE\_GROUP: I/O

PRIMITIVE\_SUBGROUP: INPUT\_BUFFER

Families: UltraScale, UltraScale+



### Introduction

The usage and rules corresponding to the differential primitives are similar to the single-ended SelectIO primitives. Differential SelectIO primitives have two pins to and from the device pads to show the P and N channel pins in a differential pair. N channel pins have a B suffix.

I/O attributes that do not impact the logic function of the component, such as IOSTANDARD, DIFF\_TERM, and IBUF\_LOW\_PWR, should be supplied to the top-level port via an appropriate property. For details on applying such properties to the associated port, see UG912: Vivado Design Suite Properties Reference Guide. Attributes that impact the functionality such as DQS\_BIAS must be supplied to the component via a generic\_map (VHDL) or parameter (Verilog) in order to have the correct simulation behavior.

### 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. Connect to top-level p-side input port.
IB	Input	1	Diff_n Buffer Input. Connect to top-level n-side input port.
O	Output	1	Buffer output

## Design Entry Method

Instantiation	Recommended
Inference	No
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
DQS_BIAS	STRING	"FALSE", "TRUE"	"FALSE"	Provides pull-up/pull-down feature required for some DQS memory interface pins or provides DC bias for certain LVDS applications.

## VHDL 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
--      UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

IBUFDS_inst : IBUFDS
generic map (
    DQS_BIAS => "FALSE" -- (FALSE, TRUE)
)
port map (
    O => O,    -- 1-bit output: Buffer output
    I => I,    -- 1-bit input: Diff_p buffer input (connect directly to top-level port)
    IB => IB  -- 1-bit input: Diff_n buffer input (connect directly to top-level port)
);

-- End of IBUFDS_inst instantiation
    
```

## Verilog Instantiation Template

```

// IBUFDS: Differential Input Buffer
//      UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

IBUFDS #(
    .DQS_BIAS("FALSE") // (FALSE, TRUE)
)
IBUFDS_inst (
    .O(O), // 1-bit output: Buffer output
    .I(I), // 1-bit input: Diff_p buffer input (connect directly to top-level port)
    .IB(IB) // 1-bit input: Diff_n buffer input (connect directly to top-level port)
);

// End of IBUFDS_inst instantiation
    
```

## For More Information

- See the *UltraScale Architecture SelectIO Resources User Guide* ([UG571](#)).
- See the [UltraScale Programmable Devices Documentation](#).

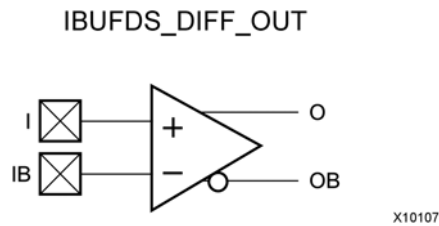
## IBUFDS\_DIFF\_OUT

Primitive: Differential Input Buffer With Complementary Outputs

PRIMITIVE\_GROUP: I/O

PRIMITIVE\_SUBGROUP: INPUT\_BUFFER

Families: UltraScale, UltraScale+



### Introduction

The IBUFDS\_DIFF\_OUT is a differential input buffer primitive with complementary outputs (O and OB).

I/O attributes that do not impact the logic function of the component, such as IOSTANDARD, DIFF\_TERM, and IBUF\_LOW\_PWR, should be supplied to the top-level port via an appropriate property. For details on applying such properties to the associated port, see UG912: Vivado Design Suite Properties Reference Guide. Attributes that impact the functionality such as DQS\_BIAS must be supplied to the component via a generic\_map (VHDL) or parameter (Verilog) in order to have the correct simulation behavior.

### 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 p-side input port.
IB	Input	1	Diff_n Buffer Input. Connect to top-level n-side input port.
O	Output	1	Buffer diff_p output
OB	Output	1	Buffer diff_n output

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
DQS_BIAS	STRING	"FALSE", "TRUE"	"FALSE"	Provides pull-up/pull-down feature required for some DQS memory interface pins or provides DC bias for certain LVDS applications.

## VHDL 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 Complementary Outputs
--                               UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

IBUFDS_DIFF_OUT_inst : IBUFDS_DIFF_OUT
generic map (
    DQS_BIAS => "FALSE" -- (FALSE, TRUE)
)
port map (
    O => O, -- 1-bit output: Buffer diff_p output
    OB => OB, -- 1-bit output: Buffer diff_n output
    I => I, -- 1-bit input: Diff_p buffer input (connect directly to top-level port)
    IB => IB -- 1-bit input: 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 Complementary Outputs
//                               UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

IBUFDS_DIFF_OUT #(
    .DQS_BIAS("FALSE") // (FALSE, TRUE)
)
IBUFDS_DIFF_OUT_inst (
    .O(O), // 1-bit output: Buffer diff_p output
    .OB(OB), // 1-bit output: Buffer diff_n output
    .I(I), // 1-bit input: Diff_p buffer input (connect directly to top-level port)
    .IB(IB) // 1-bit input: Diff_n buffer input (connect directly to top-level port)
);

// End of IBUFDS_DIFF_OUT_inst instantiation
    
```

## For More Information

- See the *UltraScale Architecture SelectIO Resources User Guide* ([UG571](#)).
- See the [UltraScale Programmable Devices Documentation](#).

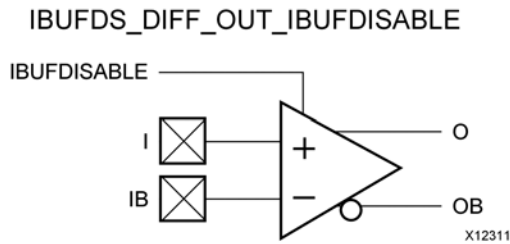
## IBUFDS\_DIFF\_OUT\_IBUFDISABLE

**Primitive: Differential Input Buffer With Complementary Outputs and Input Buffer Disable**

PRIMITIVE\_GROUP: I/O

PRIMITIVE\_SUBGROUP: INPUT\_BUFFER

Families: UltraScale, UltraScale+



### Introduction

The IBUFDS\_DIFF\_OUT\_IBUFDISABLE primitive shown is a differential input buffer with complementary differential outputs. The USE\_IBUFDISABLE attribute must be set to TRUE and the SIM\_DEVICE to ULTRASCALE for this primitive to have the expected behavior that is specific to the UltraScale architecture.

I/O attributes that do not impact the logic function of the component, such as IOSTANDARD, DIFF\_TERM, and IBUF\_LOW\_PWR, should be supplied to the top-level port via an appropriate property. For details on applying such properties to the associated port, see UG912: Vivado Design Suite Properties Reference Guide. Attributes that impact the functionality such as DQS\_BIAS must be supplied to the component via a generic\_map (VHDL) or parameter (Verilog) in order to have the correct simulation behavior.

### Port Descriptions

Port	Direction	Width	Function
I	Input	1	Diff_p Buffer Input. Connect to top-level p-side input port.
IB	Input	1	Diff_n Buffer Input. Connect to top-level n-side input port.
IBUFDISABLE	Input	1	The IBUFDISABLE feature is not supported with this primitive in the UltraScale architecture. This port must be tied to logic '0'.
O	Output	1	Buffer diff_p output
OB	Output	1	Buffer diff_n output

### Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
DQS_BIAS	STRING	"FALSE", "TRUE"	"FALSE"	Provides pull-up/pull-down feature required for some DQS memory interface pins or provides DC bias for certain LVDS applications.
SIM_DEVICE	STRING	"ULTRASCALE", "7SERIES"	"7SERIES"	This attribute must be set to "ULTRASCALE" in order to exhibit the proper simulation behavior.
USE_IBUFDISABLE	STRING	"TRUE", "FALSE"	"TRUE"	This attribute must be unspecified or set to "TRUE" if specified.

## VHDL 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_IBUFDISABLE: Differential Input Buffer With Complementary Outputs and Input Buffer Disable
--                               UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

IBUFDS_DIFF_OUT_IBUFDISABLE_inst : IBUFDS_DIFF_OUT_IBUFDISABLE
generic map (
    DQS_BIAS => "FALSE",           -- (FALSE, TRUE)
    SIM_DEVICE => "ULTRASCALE"    -- Must be set to "ULTRASCALE"
)
port map (
    O => O,                        -- 1-bit output: Buffer diff_p output
    OB => OB,                      -- 1-bit output: Buffer diff_n output
    I => I,                        -- 1-bit input: Diff_p buffer input (connect directly to top-level port)
    IB => IB,                      -- 1-bit input: Diff_n buffer input (connect directly to top-level port)
    IBUFDISABLE => IBUFDISABLE    -- 1-bit input: Must be tied to a logic '0'
);

-- End of IBUFDS_DIFF_OUT_IBUFDISABLE_inst instantiation
    
```

## Verilog Instantiation Template

```

// IBUFDS_DIFF_OUT_IBUFDISABLE: Differential Input Buffer With Complementary Outputs and Input Buffer Disable
//                               UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

IBUFDS_DIFF_OUT_IBUFDISABLE #(
    .DQS_BIAS("FALSE"),           // (FALSE, TRUE)
    .SIM_DEVICE("ULTRASCALE")    // Must be set to "ULTRASCALE"
)
IBUFDS_DIFF_OUT_IBUFDISABLE_inst (
    .O(O),                        // 1-bit output: Buffer diff_p output
    .OB(OB),                     // 1-bit output: Buffer diff_n output
    .I(I),                       // 1-bit input: Diff_p buffer input (connect directly to top-level port)
    .IB(IB),                     // 1-bit input: Diff_n buffer input (connect directly to top-level port)
    .IBUFDISABLE(IBUFDISABLE)    // 1-bit input: Must be tied to a logic '0'
);

// End of IBUFDS_DIFF_OUT_IBUFDISABLE_inst instantiation
    
```



## For More Information

- See the *UltraScale Architecture SelectIO Resources User Guide* ([UG571](#)).
- See the [UltraScale Programmable Devices Documentation](#).

## IBUFDS\_DIFF\_OUT\_INTERMDISABLE

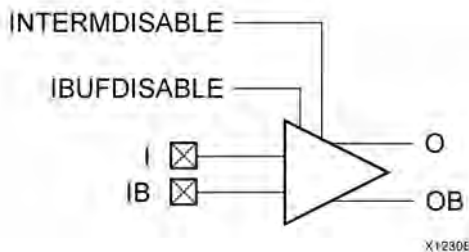
**Primitive: Differential Input Buffer with Complementary Outputs, Input Path Disable and On-die Input Termination Disable**

PRIMITIVE\_GROUP: I/O

PRIMITIVE\_SUBGROUP: INPUT\_BUFFER

Families: UltraScale, UltraScale+

IBUFDS\_DIFF\_OUT\_INTERMDISABLE



### Introduction

The IBUFDS\_DIFF\_OUT\_INTERMDISABLE primitive is available in the HR I/O banks. It has complementary differential outputs and a INTERMDISABLE port that can be used to manually disable the optional on-die receiver termination features (uncalibrated). See UG571: UltraScale Architecture SelectIO Resources, "Uncalibrated Input Termination in I/O Banks" for more details. The USE\_IBUFDISABLE attribute must be set to TRUE and SIM\_DEVICE to ULTRASCALE for this primitive to have the expected behavior that is specific to the UltraScale architecture.

If the I/O is using any on-die receiver termination features (uncalibrated), this primitive disables the termination legs whenever the INTERMDISABLE signal is asserted High.

I/O attributes that do not impact the logic function of the component, such as IOSTANDARD, DIFF\_TERM, and IBUF\_LOW\_PWR, should be supplied to the top-level port via an appropriate property. For details on applying such properties to the associated port, see UG912: Vivado Design Suite Properties Reference Guide. Attributes that impact the functionality such as DQS\_BIAS must be supplied to the component via a generic\_map (VHDL) or parameter (Verilog) in order to have the correct simulation behavior.

### Port Descriptions

Port	Direction	Width	Function
I	Input	1	Diff_p Buffer Input. Connect to top-level p-side input port.
IB	Input	1	Diff_n Buffer Input. Connect to top-level n-side input port.
IBUFDISABLE	Input	1	The IBUFDISABLE feature is not supported with this primitive in the UltraScale architecture. This port must be tied to logic '0'.
INTERMDISABLE	Input	1	Disables input termination reducing current dissipation within the buffer. This feature is generally used to reduce power at times when the I/O is idle for a period of time.
O	Output	1	Buffer diff_p output

Port	Direction	Width	Function
OB	Output	1	Buffer diff_n output

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
DQS_BIAS	STRING	"FALSE", "TRUE"	"FALSE"	Provides pull-up/pull-down feature required for some DQS memory interface pins or provides DC bias for certain LVDS applications.
SIM_DEVICE	STRING	"ULTRASCALE", "7SERIES"	"7SERIES"	This attribute must be set to "ULTRASCALE" in order to exhibit the proper simulation behavior.
USE_IBUFDISABLE	STRING	"TRUE", "FALSE"	"TRUE"	This attribute must be unspecified or set to "TRUE" if specified.

## VHDL 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_INTERMDISABLE: Differential Input Buffer with Complementary Outputs, Input Path Disable and On-die Input Termination Disable
--                                     UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

IBUFDS_DIFF_OUT_INTERMDISABLE_inst : IBUFDS_DIFF_OUT_INTERMDISABLE
generic map (
    DQS_BIAS => "FALSE",           -- (FALSE, TRUE)
    SIM_DEVICE => "ULTRASCALE"    -- Must be set to "ULTRASCALE"
)
port map (
    O => O,                        -- 1-bit output: Buffer diff_p output
    OB => OB,                      -- 1-bit output: Buffer diff_n output
    I => I,                        -- 1-bit input: Diff_p buffer input (connect directly to top-level port)
    IB => IB,                      -- 1-bit input: Diff_n buffer input (connect directly to top-level port)
    IBUFDISABLE => IBUFDISABLE,    -- 1-bit input: Must be tied to a logic '0'
    INTERMDISABLE => INTERMDISABLE -- 1-bit input: Buffer termination disable, high=disable
);

-- End of IBUFDS_DIFF_OUT_INTERMDISABLE_inst instantiation
    
```

## Verilog Instantiation Template

```
// IBUFDS_DIFF_OUT_INTERMDISABLE: Differential Input Buffer with Complementary Outputs, Input Path Disable and On-die Input Termination Disable
//                                     UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

IBUFDS_DIFF_OUT_INTERMDISABLE #(
    .DQS_BIAS("FALSE"), // (FALSE, TRUE)
    .SIM_DEVICE("ULTRASCALE") // Must be set to "ULTRASCALE"
)
IBUFDS_DIFF_OUT_INTERMDISABLE_inst (
    .O(O), // 1-bit output: Buffer diff_p output
    .OB(OB), // 1-bit output: Buffer diff_n output
    .I(I), // 1-bit input: Diff_p buffer input (connect directly to top-level port)
    .IB(IB), // 1-bit input: Diff_n buffer input (connect directly to top-level port)
    .IBUFDISABLE(IBUFDISABLE), // 1-bit input: Must be tied to a logic '0'
    .INTERMDISABLE(INTERMDISABLE) // 1-bit input: Buffer termination disable, high=disable
);

// End of IBUFDS_DIFF_OUT_INTERMDISABLE_inst instantiation
```

## For More Information

- See the *UltraScale Architecture SelectIO Resources User Guide* ([UG571](#)).
- See the [UltraScale Programmable Devices Documentation](#).

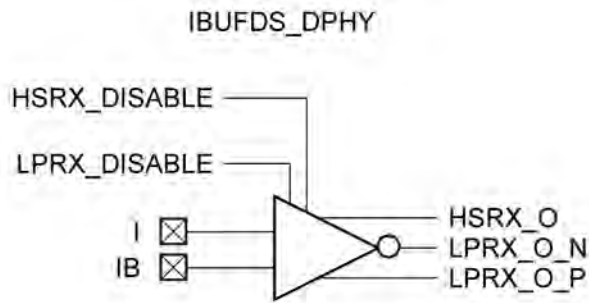
# IBUFDS\_DPHY

Primitive: Differential Input Buffer with MIPI support

PRIMITIVE\_GROUP: I/O

PRIMITIVE\_SUBGROUP: INPUT\_BUFFER

Families: UltraScale+



X15113-101115

## Introduction

Differential Input Buffer with MIPI support

## Port Descriptions

Port	Direction	Width	Function
HSRX_DISABLE	Input	1	Disable control for HS mode
HSRX_O	Output	1	HS RX output
I	Input	1	Data input0 PAD
IB	Input	1	Data input1 PAD
LPRX_DISABLE	Input	1	Disable control for LP mode
LPRX_O_N	Output	1	LP RX output (Slave)
LPRX_O_P	Output	1	LP RX output (Master)

## Design Entry Method

Instantiation	Recommended
Inference	No
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
DIFF_TERM	BOOLEAN	TRUE, FALSE	TRUE	Turns the built-in differential termination on (TRUE) or off (FALSE)
IOSTANDARD	STRING	String	"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_DPHY: Differential Input Buffer with MIPI support
-- UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

IBUFDS_DPHY_inst : IBUFDS_DPHY
generic map (
    DIFF_TERM => TRUE,          -- Differential termination
    IOSTANDARD => "DEFAULT"    -- I/O standard
)
port map (
    HSRX_O => HSRX_O,          -- 1-bit output: HS RX output
    LPRX_O_N => LPRX_O_N,      -- 1-bit output: LP RX output (Slave)
    LPRX_O_P => LPRX_O_P,      -- 1-bit output: LP RX output (Master)
    HSRX_DISABLE => HSRX_DISABLE, -- 1-bit input: Disable control for HS mode
    I => I,                    -- 1-bit input: Data input0 PAD
    IB => IB,                  -- 1-bit input: Data input1 PAD
    LPRX_DISABLE => LPRX_DISABLE -- 1-bit input: Disable control for LP mode
);

-- End of IBUFDS_DPHY_inst instantiation
    
```

## Verilog Instantiation Template

```

// IBUFDS_DPHY: Differential Input Buffer with MIPI support
// UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

IBUFDS_DPHY #(
    .DIFF_TERM("TRUE"),        // Differential termination
    .IOSTANDARD("DEFAULT")    // I/O standard
)
IBUFDS_DPHY_inst (
    .HSRX_O(HSRX_O),          // 1-bit output: HS RX output
    .LPRX_O_N(LPRX_O_N),      // 1-bit output: LP RX output (Slave)
    .LPRX_O_P(LPRX_O_P),      // 1-bit output: LP RX output (Master)
    .HSRX_DISABLE(HSRX_DISABLE), // 1-bit input: Disable control for HS mode
    .I(I),                    // 1-bit input: Data input0 PAD
    .IB(IB),                  // 1-bit input: Data input1 PAD
    .LPRX_DISABLE(LPRX_DISABLE) // 1-bit input: Disable control for LP mode
);

// End of IBUFDS_DPHY_inst instantiation
    
```

## For More Information

- See the *UltraScale Architecture SelectIO Resources User Guide* ([UG571](#)).
- See the [UltraScale Programmable Devices Documentation](#).

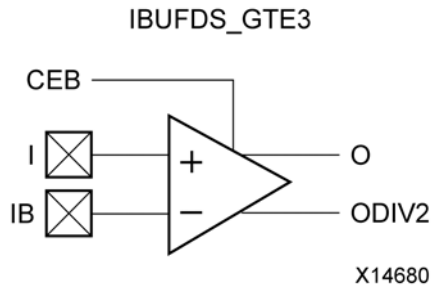
## IBUFDS\_GTE3

Primitive: Gigabit Transceiver Buffer

PRIMITIVE\_GROUP: ADVANCED

PRIMITIVE\_SUBGROUP: GT

Families: UltraScale



### Introduction

IBUFDS\_GTE3 is the gigabit transceiver input pad buffer component. The REFCLK signal should be routed to the dedicated reference clock input pins on the serial transceiver, and the user design should instantiate the IBUFDS\_GTE3 primitive in the user design. See the Transceivers User Guide for more information on PCB layout requirements, including reference clock requirements.

### Port Descriptions

Port	Direction	Width	Function
CEB	Input	1	Refer to Transceiver User Guide
I	Input	1	Refer to Transceiver User Guide
IB	Input	1	Refer to Transceiver User Guide
O	Output	1	Refer to Transceiver User Guide
ODIV2	Output	1	Refer to Transceiver User Guide

### Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	Recommended



## Available Attributes

Attribute	Type	Allowed Values	Default	Description
REFCLK_EN_TX_PATH	BINARY	1'b0 to 1'b1	1'b0	Refer to Transceiver User Guide
REFCLK_HROW_CK_SEL	BINARY	2'b00 to 2'b11	2'b00	Refer to Transceiver User Guide
REFCLK_ICNTL_RX	BINARY	2'b00 to 2'b11	2'b00	Refer to Transceiver User Guide

## VHDL 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_GTE3: Gigabit Transceiver Buffer
--           UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

IBUFDS_GTE3_inst : IBUFDS_GTE3
generic map (
    REFCLK_EN_TX_PATH => '0', -- Refer to Transceiver User Guide
    REFCLK_HROW_CK_SEL => "00", -- Refer to Transceiver User Guide
    REFCLK_ICNTL_RX => "00" -- Refer to Transceiver User Guide
)
port map (
    O => O, -- 1-bit output: Refer to Transceiver User Guide
    ODIV2 => ODIV2, -- 1-bit output: Refer to Transceiver User Guide
    CEB => CEB, -- 1-bit input: Refer to Transceiver User Guide
    I => I, -- 1-bit input: Refer to Transceiver User Guide
    IB => IB -- 1-bit input: Refer to Transceiver User Guide
);

-- End of IBUFDS_GTE3_inst instantiation
    
```

## Verilog Instantiation Template

```

// IBUFDS_GTE3: Gigabit Transceiver Buffer
//           UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

IBUFDS_GTE3 #(
    .REFCLK_EN_TX_PATH(1'b0), // Refer to Transceiver User Guide
    .REFCLK_HROW_CK_SEL(2'b00), // Refer to Transceiver User Guide
    .REFCLK_ICNTL_RX(2'b00) // Refer to Transceiver User Guide
)
IBUFDS_GTE3_inst (
    .O(O), // 1-bit output: Refer to Transceiver User Guide
    .ODIV2(ODIV2), // 1-bit output: Refer to Transceiver User Guide
    .CEB(CEB), // 1-bit input: Refer to Transceiver User Guide
    .I(I), // 1-bit input: Refer to Transceiver User Guide
    .IB(IB) // 1-bit input: Refer to Transceiver User Guide
);

// End of IBUFDS_GTE3_inst instantiation
    
```

## For More Information

- See the *UltraScale Architecture GTH Transceivers User Guide* ([UG576](#))
- See the *UltraScale Architecture GTY Transceivers Advance Specification User Guide* ([UG578](#))
- See the [UltraScale Programmable Devices Documentation](#).

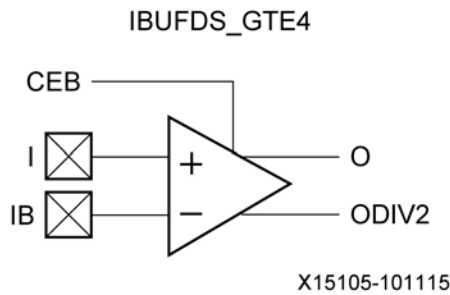
## IBUFDS\_GTE4

Primitive: Gigabit Transceiver Buffer

PRIMITIVE\_GROUP: ADVANCED

PRIMITIVE\_SUBGROUP: GT

Families: UltraScale+



### Introduction

IBUFDS\_GTE4 is the gigabit transceiver input pad buffer component. The REFCLK signal should be routed to the dedicated reference clock input pins on the serial transceiver, and the user design should instantiate the IBUFDS\_GTE4 primitive in the user design. See the Transceivers User Guide for more information on PCB layout requirements, including reference clock requirements.

### Port Descriptions

Port	Direction	Width	Function
CEB	Input	1	Refer to Transceiver User Guide
I	Input	1	Refer to Transceiver User Guide
IB	Input	1	Refer to Transceiver User Guide
O	Output	1	Refer to Transceiver User Guide
ODIV2	Output	1	Refer to Transceiver User Guide

### Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	Recommended

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
REFCLK_EN_TX_PATH	BINARY	1'b0 to 1'b1	1'b0	Refer to Transceiver User Guide
REFCLK_HROW_CK_SEL	BINARY	2'b00 to 2'b11	2'b00	Refer to Transceiver User Guide
REFCLK_ICNTL_RX	BINARY	2'b00 to 2'b11	2'b00	Refer to Transceiver User Guide

## VHDL 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_GTE4: Gigabit Transceiver Buffer
-- UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

IBUFDS_GTE4_inst : IBUFDS_GTE4
generic map (
    REFCLK_EN_TX_PATH => '0', -- Refer to Transceiver User Guide
    REFCLK_HROW_CK_SEL => "00", -- Refer to Transceiver User Guide
    REFCLK_ICNTL_RX => "00" -- Refer to Transceiver User Guide
)
port map (
    O => O, -- 1-bit output: Refer to Transceiver User Guide
    ODIV2 => ODIV2, -- 1-bit output: Refer to Transceiver User Guide
    CEB => CEB, -- 1-bit input: Refer to Transceiver User Guide
    I => I, -- 1-bit input: Refer to Transceiver User Guide
    IB => IB -- 1-bit input: Refer to Transceiver User Guide
);

-- End of IBUFDS_GTE4_inst instantiation
    
```

## Verilog Instantiation Template

```

// IBUFDS_GTE4: Gigabit Transceiver Buffer
// UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

IBUFDS_GTE4 #(
    .REFCLK_EN_TX_PATH(1'b0), // Refer to Transceiver User Guide
    .REFCLK_HROW_CK_SEL(2'b00), // Refer to Transceiver User Guide
    .REFCLK_ICNTL_RX(2'b00) // Refer to Transceiver User Guide
)
IBUFDS_GTE4_inst (
    .O(O), // 1-bit output: Refer to Transceiver User Guide
    .ODIV2(ODIV2), // 1-bit output: Refer to Transceiver User Guide
    .CEB(CEB), // 1-bit input: Refer to Transceiver User Guide
    .I(I), // 1-bit input: Refer to Transceiver User Guide
    .IB(IB) // 1-bit input: Refer to Transceiver User Guide
);

// End of IBUFDS_GTE4_inst instantiation
    
```

## For More Information

- See the *UltraScale Architecture GTH Transceivers User Guide* ([UG576](#))
- See the *UltraScale Architecture GTY Transceivers Advance Specification User Guide* ([UG578](#))
- See the [UltraScale Programmable Devices Documentation](#).

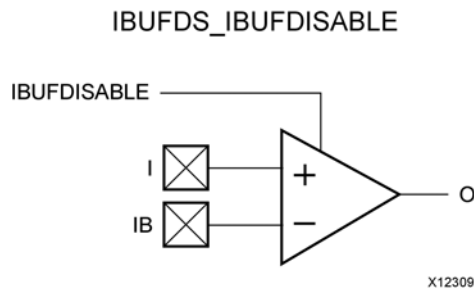
## IBUFDS\_IBUFDISABLE

**Primitive: Differential Input Buffer With Input Buffer Disable**

PRIMITIVE\_GROUP: I/O

PRIMITIVE\_SUBGROUP: INPUT\_BUFFER

Families: UltraScale, UltraScale+



### Introduction

This primitive is a differential input buffer with input disable for additional power savings when the input data is not needed. The `USE_IBUFDISABLE` attribute must be set to `TRUE` and the `SIM_DEVICE` to `ULTRASCALE` for this primitive to have the expected UltraScale architecture specific behavior.

I/O attributes that do not impact the logic function of the component, such as `IOSTANDARD`, `DIFF_TERM`, and `IBUF_LOW_PWR`, should be supplied to the top-level port via an appropriate property. For details on applying such properties to the associated port, see UG912: Vivado Design Suite Properties Reference Guide. Attributes that impact the functionality such as `DQS_BIAS` must be supplied to the component via a `generic_map` (VHDL) or `parameter` (Verilog) in order to have the correct simulation behavior.

### Port Descriptions

Port	Direction	Width	Function
I	Input	1	Diff_p Buffer Input. Connect to top-level p-side input port.
IB	Input	1	Diff_n Buffer Input. Connect to top-level n-side input port.
IBUFDISABLE	Input	1	The IBUFDISABLE feature is not supported with this primitive in the UltraScale architecture. This port must be tied to logic '0'.
O	Output	1	Buffer output

### Design Entry Method

Instantiation	Recommended
Inference	No
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
DQS_BIAS	STRING	"FALSE", "TRUE"	"FALSE"	Provides pull-up/pull-down feature required for some DQS memory interface pins or provides DC bias for certain LVDS applications.
SIM_DEVICE	STRING	"ULTRASCALE", "7SERIES"	"7SERIES"	This attribute must be set to "ULTRASCALE" in order to exhibit the proper simulation behavior.
USE_IBUFDISABLE	STRING	"TRUE", "FALSE"	"TRUE"	This attribute must be unspecified or set to "TRUE" if specified.

## VHDL 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_IBUFDISABLE: Differential Input Buffer With Input Buffer Disable
--                               UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

IBUFDS_IBUFDISABLE_inst : IBUFDS_IBUFDISABLE
generic map (
    DQS_BIAS => "FALSE",           -- (FALSE, TRUE)
    SIM_DEVICE => "ULTRASCALE"    -- Must be set to "ULTRASCALE"
)
port map (
    O => O,                        -- 1-bit output: Buffer output
    I => I,                        -- 1-bit input: Diff_p buffer input (connect directly to top-level port)
    IB => IB,                      -- 1-bit input: Diff_n buffer input (connect directly to top-level port)
    IBUFDISABLE => IBUFDISABLE    -- 1-bit input: Must be tied to a logic '0'
);

-- End of IBUFDS_IBUFDISABLE_inst instantiation
    
```

## Verilog Instantiation Template

```

// IBUFDS_IBUFDISABLE: Differential Input Buffer With Input Buffer Disable
//                               UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

IBUFDS_IBUFDISABLE #(
    .DQS_BIAS("FALSE"),           // (FALSE, TRUE)
    .SIM_DEVICE("ULTRASCALE")    // Must be set to "ULTRASCALE"
)
IBUFDS_IBUFDISABLE_inst (
    .O(O),                        // 1-bit output: Buffer output
    .I(I),                        // 1-bit input: Diff_p buffer input (connect directly to top-level port)
    .IB(IB),                      // 1-bit input: Diff_n buffer input (connect directly to top-level port)
    .IBUFDISABLE(IBUFDISABLE)    // 1-bit input: Must be tied to a logic '0'
);

// End of IBUFDS_IBUFDISABLE_inst instantiation
    
```

## For More Information

- See the *UltraScale Architecture SelectIO Resources User Guide* ([UG571](#)).
- See the [UltraScale Programmable Devices Documentation](#).



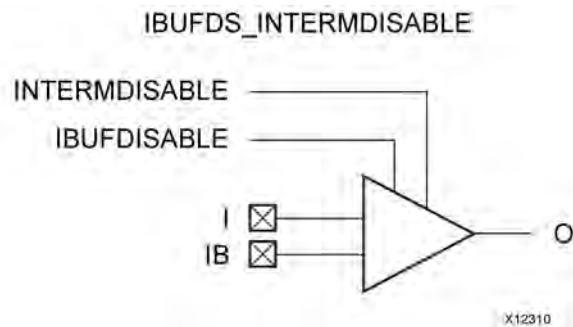
## IBUFDS\_INTERMDISABLE

**Primitive: Differential Input Buffer With Input Buffer Disable and On-die Input Termination Disable**

PRIMITIVE\_GROUP: I/O

PRIMITIVE\_SUBGROUP: INPUT\_BUFFER

Families: UltraScale, UltraScale+



### Introduction

The IBUFDS\_INTERMDISABLE primitive is available in the HR I/O banks, is similar to the IBUFDS\_IBUFDISABLE primitive because it has a IBUFDISABLE port to disable the input buffer when not in use. The IBUFDS\_INTERMDISABLE primitive also has an INTERMDISABLE port to use to disable the optional on-die receiver termination feature. See UG571: UltraScale Architecture SelectIO Resources, "Uncalibrated Input Termination in I/O Banks" for more details.

The IBUFDS\_INTERMDISABLE primitive can disable the input buffer and force the O output to a logic-Low when the IBUFDISABLE signal is asserted High. The USE\_IBUFDISABLE attribute must be set to TRUE and SIM\_DEVICE to ULTRASCALE for this primitive to have the expected behavior that is specific to the UltraScale architecture. If the I/O is using the optional on-die receiver termination feature, this primitive disables the termination legs whenever the INTERMDISABLE signal is asserted High. Both these features can be combined to reduce power whenever the input is idle.

I/O attributes that do not impact the logic function of the component, such as IOSTANDARD, DIFF\_TERM, and IBUF\_LOW\_PWR, should be supplied to the top-level port via an appropriate property. For details on applying such properties to the associated port, see UG912: Vivado Design Suite Properties Reference Guide. Attributes that impact the functionality, such as DQS\_BIAS, must be supplied to the component via a generic\_map (VHDL) or parameter (Verilog) in order to have the correct simulation behavior.

## Port Descriptions

Port	Direction	Width	Function
I	Input	1	Diff_p Buffer Input. Connect to top-level p-side input port.
IB	Input	1	Diff_n Buffer Input. Connect to top-level n-side input port.
IBUFDISABLE	Input	1	Disables input path through the buffer and forces to a logic low. This feature is generally used to reduce power at times when the I/O is idle for a period of time.
INTERMDISABLE	Input	1	Disables input termination reducing current dissipation within the buffer. This feature is generally used to reduce power at times when the I/O is idle for a period of time.
O	Output	1	Buffer output

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
DQS_BIAS	STRING	"FALSE", "TRUE"	"FALSE"	Provides pull-up/pull-down feature required for some DQS memory interface pins or provides DC bias for certain LVDS applications.
SIM_DEVICE	STRING	"ULTRASCALE", "7SERIES"	"7SERIES"	This attribute must be set to "ULTRASCALE" in order to exhibit the proper simulation behavior.
USE_IBUFDISABLE	STRING	"TRUE", "FALSE"	"TRUE"	This attribute must be unspecified or set to "TRUE" if specified.

## VHDL 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_INTERMDISABLE: Differential Input Buffer With Input Buffer Disable and On-die Input Termination Disable
--                               UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

IBUFDS_INTERMDISABLE_inst : IBUFDS_INTERMDISABLE
generic map (
    DQS_BIAS => "FALSE",           -- (FALSE, TRUE)
    SIM_DEVICE => "ULTRASCALE"    -- Must be set to "ULTRASCALE"
)
port map (
    O => O,                       -- 1-bit output: Buffer output
    I => I,                       -- 1-bit input: Diff_p buffer input (connect directly to top-level port)

```

```

IB => IB,                -- 1-bit input: Diff_n buffer input (connect directly to top-level port)
IBUFDISABLE => IBUFDISABLE,    -- 1-bit input: Buffer input disable, high=disable
INTERMDISABLE => INTERMDISABLE -- 1-bit input: Buffer termination disable, high=disable
);

-- End of IBUFDS_INTERMDISABLE_inst instantiation
    
```

## Verilog Instantiation Template

```

// IBUFDS_INTERMDISABLE: Differential Input Buffer With Input Buffer Disable and On-die Input Termination Disable
//                               UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

IBUFDS_INTERMDISABLE #(
    .DQS_BIAS("FALSE"),        // (FALSE, TRUE)
    .SIM_DEVICE("ULTRASCALE") // Must be set to "ULTRASCALE"
)
IBUFDS_INTERMDISABLE_inst (
    .O(O),                    // 1-bit output: Buffer output
    .I(I),                    // 1-bit input: Diff_p buffer input (connect directly to top-level port)
    .IB(IB),                  // 1-bit input: Diff_n buffer input (connect directly to top-level port)
    .IBUFDISABLE(IBUFDISABLE), // 1-bit input: Buffer input disable, high=disable
    .INTERMDISABLE(INTERMDISABLE) // 1-bit input: Buffer termination disable, high=disable
);

// End of IBUFDS_INTERMDISABLE_inst instantiation
    
```

## For More Information

- See the *UltraScale Architecture SelectIO Resources User Guide* ([UG571](#)).
- See the [UltraScale Programmable Devices Documentation](#).

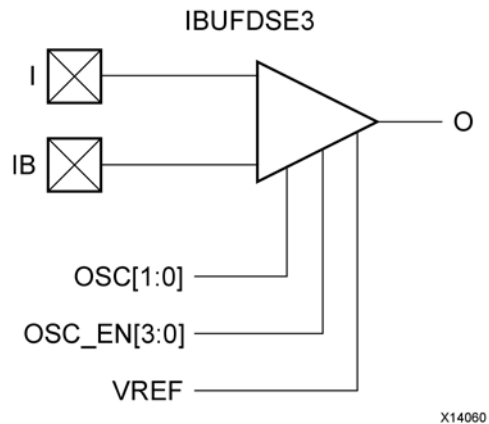
## IBUFDSE3

Primitive: Differential Input Buffer with Offset Calibration

PRIMITIVE\_GROUP: I/O

PRIMITIVE\_SUBGROUP: INPUT\_BUFFER

Families: UltraScale, UltraScale+



### Introduction

The differential input buffer (IBUFDSE3) primitive is only supported in HP I/O banks. This UltraScale architecture specific primitive has functions similar to the IBUFDSE3\_IBUFDISABLE along with controls for offset calibration and input buffer disable (IBUFDISABLE). The offset calibration feature is accessed using the OSC\_EN and OSC[3:0] ports. The V<sub>REF</sub> scan feature is not supported with this primitive.

I/O attributes that do not impact the logic function of the component, such as IOSTANDARD, DIFF\_TERM, and IBUF\_LOW\_PWR, should be supplied in XDC or to the top-level port via an appropriate property. For details on applying such properties to XDC or the associated port, see UG912: Vivado Design Suite Properties Reference Guide. Attributes that impact the functionality such as DQS\_BIAS must be supplied to the component via a generic\_map (VHDL) or parameter (Verilog) in order to have the correct simulation behavior.

### Port Descriptions

Port	Direction	Width	Function
I	Input	1	Diff_p Buffer Input. Connect to top-level p-side input port.
IB	Input	1	Diff_n Buffer Input. Connect to top-level n-side input port.
IBUFDISABLE	Input	1	Disables input path through the buffer and forces to a logic low. This feature is generally used to reduce power at times when the I/O is idle for a period of time.
O	Output	1	Buffer output
OSC<3:0>	Input	4	Offset cancellation value
OSC_EN<1:0>	Input	2	Offset cancellation enable

## Design Entry Method

Instantiation	Recommended
Inference	No
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
DQS_BIAS	STRING	"FALSE", "TRUE"	"FALSE"	Provides pull-up/pull-down feature required for some DQS memory interface pins or provides DC bias for certain LVDS applications.
SIM_INPUT_BUFFER_OFFSET	DECIMAL	-50 to 50	0	Offset value for simulation purposes.
USE_IBUFDISABLE	STRING	"FALSE", "TRUE"	"FALSE"	This attribute must be unspecified or set to "TRUE" if specified.

## VHDL Instantiation Template

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- IBUFDSE3: Differential Input Buffer with Offset Calibration
--           UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

IBUFDSE3_inst : IBUFDSE3
generic map (
    DQS_BIAS => "FALSE",           -- (FALSE, TRUE)
    SIM_INPUT_BUFFER_OFFSET => 0  -- Offset value for simulation (-50-50)
)
port map (
    O => O,                        -- 1-bit output: Buffer output
    I => I,                        -- 1-bit input: Diff_p buffer input (connect directly to top-level port)
    IB => IB,                      -- 1-bit input: Diff_n buffer input (connect directly to top-level port)
    IBUFDISABLE => IBUFDISABLE,    -- 1-bit input: Buffer disable input, high=disable
    OSC => OSC,                   -- 4-bit input: Offset cancellation value
    OSC_EN => OSC_EN              -- 2-bit input: Offset cancellation enable
);

-- End of IBUFDSE3_inst instantiation
    
```

## Verilog Instantiation Template

```
// IBUFDSE3: Differential Input Buffer with Offset Calibration
// UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

IBUFDSE3 #(
    .DQS_BIAS("FALSE"),           // (FALSE, TRUE)
    .SIM_INPUT_BUFFER_OFFSET(0)   // Offset value for simulation (-50-50)
)
IBUFDSE3_inst (
    .O(O),                        // 1-bit output: Buffer output
    .I(I),                        // 1-bit input: Diff_p buffer input (connect directly to top-level port)
    .IB(IB),                      // 1-bit input: Diff_n buffer input (connect directly to top-level port)
    .IBUFDISABLE(IBUFDISABLE),    // 1-bit input: Buffer disable input, high=disable
    .OSC(OSC),                    // 4-bit input: Offset cancellation value
    .OSC_EN(OSC_EN)              // 2-bit input: Offset cancellation enable
);

// End of IBUFDSE3_inst instantiation
```

## For More Information

- See the *UltraScale Architecture SelectIO Resources User Guide* ([UG571](#)).
- See the [UltraScale Programmable Devices Documentation](#).

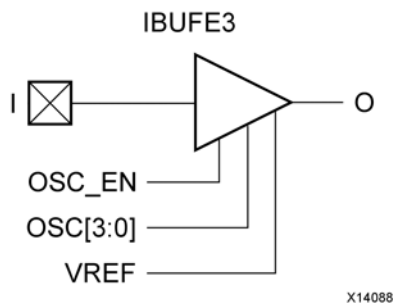
## IBUFE3

Primitive: Input Buffer with Offset Calibration and VREF Tuning

PRIMITIVE\_GROUP: I/O

PRIMITIVE\_SUBGROUP: INPUT\_BUFFER

Families: UltraScale, UltraScale+



### Introduction

The input buffer (IBUFE3) primitive is only supported in HP I/O banks. This UltraScale architecture specific primitive has functions similar to the IBUF\_IBUFDISABLE with added controls for offset calibration and  $V_{REF}$  tuning, along with input buffer disable (IBUFDISABLE). The offset calibration feature is accessed using the OSC\_EN and OSC[3:0] ports. The  $V_{REF}$  scan feature is accessed using the HPIO\_VREF primitive in conjunction with IBUFE3.

I/O attributes that do not impact the logic function of the component, such as IOSTANDARD, and IBUF\_LOW\_PWR, should be supplied in XDC or to the top-level port via an appropriate property. For details on applying such properties to the associated port, see UG912: Vivado Design Suite Properties Reference Guide. Attributes that impact the functionality, such as SIM\_INPUT\_BUFFER\_OFFSET, must be supplied to the component via a generic\_map (VHDL) or parameter (Verilog) in order to have the correct simulation behavior.

### Port Descriptions

Port	Direction	Width	Function
I	Input	1	Diff_p Buffer Input. Connect to top-level p-side input port.
IBUFDISABLE	Input	1	Disables input path through the buffer and forces to a logic low. This feature is generally used to reduce power at times when the I/O is idle for a period of time.
O	Output	1	Buffer output
OSC<3:0>	Input	4	Offset cancellation value
OSC_EN	Input	1	Offset cancellation enable
VREF	Input	1	Vref input from HPIO_VREF

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
SIM_INPUT_BUFFER_OFFSET	DECIMAL	-50 to 50	0	Offset value for simulation purposes.
USE_IBUFDISABLE	STRING	"FALSE", "TRUE"	"FALSE"	This attribute must be unspecified or set to "TRUE" if specified.

## VHDL Instantiation Template

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- IBUFE3: Input Buffer with Offset Calibration and VREF Tuning
-- UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

IBUFE3_inst : IBUFE3
generic map (
    SIM_INPUT_BUFFER_OFFSET => 0 -- Offset value for simulation (-50-50)
)
port map (
    O => O,                -- 1-bit output: Buffer output
    I => I,                -- 1-bit input: Buffer input (connect directly to top-level port)
    IBUFDISABLE => IBUFDISABLE, -- 1-bit input: Buffer disable input, high=disable
    OSC => OSC,            -- 4-bit input: Offset cancellation value
    OSC_EN => OSC_EN,     -- 1-bit input: Offset cancellation enable
    VREF => VREF           -- 1-bit input: Vref input from HPIO_VREF
);

-- End of IBUFE3_inst instantiation
    
```

## Verilog Instantiation Template

```

// IBUFE3: Input Buffer with Offset Calibration and VREF Tuning
// UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

IBUFE3 #(
    .SIM_INPUT_BUFFER_OFFSET(0) // Offset value for simulation (-50-50)
)
IBUFE3_inst (
    .O(O),                // 1-bit output: Buffer output
    .I(I),                // 1-bit input: Buffer input (connect directly to top-level port)
    .IBUFDISABLE(IBUFDISABLE), // 1-bit input: Buffer disable input, high=disable
    .OSC(OSC),            // 4-bit input: Offset cancellation value
    .OSC_EN(OSC_EN),     // 1-bit input: Offset cancellation enable
    .VREF(VREF)           // 1-bit input: Vref input from HPIO_VREF
);
    
```



```
// End of IBUFE3_inst instantiation
```

## For More Information

- See the *UltraScale Architecture SelectIO Resources User Guide* ([UG571](#)).
- See the [UltraScale Programmable Devices Documentation](#).

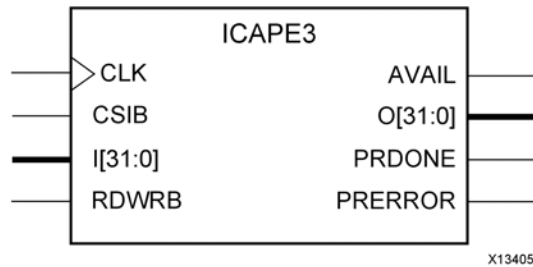
# ICAPE3

## Primitive: Internal Configuration Access Port

PRIMITIVE\_GROUP: CONFIGURATION

PRIMITIVE\_SUBGROUP: ICAP

Families: UltraScale, UltraScale+



## Introduction

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

## Port Descriptions

Port	Direction	Width	Function
AVAIL	Output	1	Availability status of ICAP
CLK	Input	1	Clock input
CSIB	Input	1	Active-Low ICAP enable
I<31:0>	Input	32	Configuration data input bus
O<31:0>	Output	32	Configuration data output bus
PRDONE	Output	1	Indicates completion of Partial Reconfiguration
PRERROR	Output	1	Indicates Error during Partial Reconfiguration
RDWRB	Input	1	Read/Write Select input

## Design Entry Method

Instantiation	Recommended
Inference	No
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
DEVICE_ID	HEX	32'h03628093, 32'h03627093	32'h03628093	Specifies the pre-programmed Device ID value to be used for simulation purposes.
ICAP_AUTO_SWITCH	STRING	"DISABLE", "ENABLE"	"DISABLE"	Enable switch ICAP using sync word
SIM_CFG_FILE_NAME	STRING	String	"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;

-- ICAPE3: Internal Configuration Access Port
--      UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

ICAPE3_inst : ICAPE3
generic map (
    DEVICE_ID => X"03628093",      -- Specifies the pre-programmed Device ID value to be used for simulation
                                   -- purposes.
    ICAP_AUTO_SWITCH => "DISABLE", -- Enable switch ICAP using sync word
    SIM_CFG_FILE_NAME => "NONE"   -- Specifies the Raw Bitstream (RBT) file to be parsed by the simulation
                                   -- model
)
port map (
    AVAIL => AVAIL,      -- 1-bit output: Availability status of ICAP
    O => O,              -- 32-bit output: Configuration data output bus
    PRDONE => PRDONE,   -- 1-bit output: Indicates completion of Partial Reconfiguration
    PRERROR => PRERROR, -- 1-bit output: Indicates Error during Partial Reconfiguration
    CLK => CLK,         -- 1-bit input: Clock input
    CSIB => CSIB,       -- 1-bit input: Active-Low ICAP enable
    I => I,              -- 32-bit input: Configuration data input bus
    RDWRB => RDWRB      -- 1-bit input: Read/Write Select input
);

-- End of ICAPE3_inst instantiation
    
```

## Verilog Instantiation Template

```

// ICAPE3: Internal Configuration Access Port
//      UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

ICAPE3 #(
    .DEVICE_ID(32'h03628093),      // Specifies the pre-programmed Device ID value to be used for simulation
                                   // purposes.
    .ICAP_AUTO_SWITCH("DISABLE"), // Enable switch ICAP using sync word
    .SIM_CFG_FILE_NAME("NONE")   // Specifies the Raw Bitstream (RBT) file to be parsed by the simulation
                                   // model
)
ICAPE3_inst (
    .AVAIL(AVAIL),      // 1-bit output: Availability status of ICAP
    .O(O),              // 32-bit output: Configuration data output bus
    .PRDONE(PRDONE),   // 1-bit output: Indicates completion of Partial Reconfiguration
    
```

```
.PRERROR(PRERROR), // 1-bit output: Indicates Error during Partial Reconfiguration
.CLK(CLK),          // 1-bit input: Clock input
.CSIB(CSIB),        // 1-bit input: Active-Low ICAP enable
.I(I),              // 32-bit input: Configuration data input bus
.RDWRB(RDWRB)       // 1-bit input: Read/Write Select input
);

// End of ICAPE3_inst instantiation
```

## For More Information

- See the *UltraScale Architecture Configuration User Guide* ([UG570](#)).
- See the [UltraScale Programmable Devices Documentation](#).

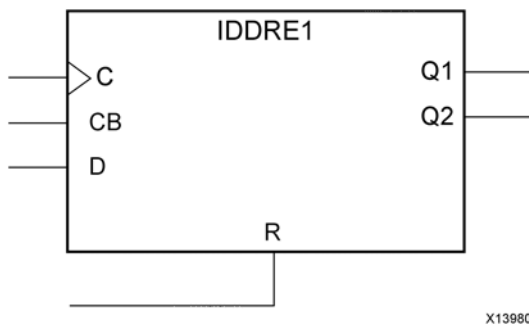
## IDDRE1

Primitive: Dedicated Dual Data Rate (DDR) Input Register

PRIMITIVE\_GROUP: REGISTER

PRIMITIVE\_SUBGROUP: DDR

Families: UltraScale, UltraScale+



### Introduction

In component mode, the IDDRE1 in UltraScale devices is a dedicated input register designed to receive external dual data rate (DDR) signals into Xilinx devices. The IDDRE1 is available with modes that present the data to the device 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.

### Port Descriptions

Port	Direction	Width	Function
C	Input	1	The high-speed clock input (C) is used to clock in the input serial data stream
CB	Input	1	The inverted high-speed clock input
D	Input	1	The serial input data port (D) is the serial (high-speed) data input port of the ISERDESE3. This port accepts data from the IOB or device Fabric.
Q1	Output	1	Registered parallel output 1
Q2	Output	1	Registered parallel output 2
R	Input	1	Active High Asynchronous Reset

### Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	Yes

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
DDR_CLK_EDGE	STRING	"OPPOSITE_EDGE", "SAME_EDGE", "SAME_EDGE_PIPELINED"	"OPPOSITE_EDGE"	Sets the IDDRE1 mode of operation with respect to clock edge. <ul style="list-style-type: none"> <li>"OPPOSITE_EDGE": Traditional input DDR solution. Data presented to Q1 on the rising edge and Q2 on the falling edge.</li> <li>"SAME_EDGE": Data is presented to the device logic on the same clock edge. Has separated effect.</li> <li>"SAME_EDGE_PIPELINED": Data is presented to the device logic on the same clock edge. Removes the separated effect but incurs clock latency.</li> </ul>
IS_C_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether the clock C pin is active-high or active-low.
IS_CB_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether the clock CB pin is active-high or active-low.

## VHDL Instantiation Template

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- IDDRE1: Dedicated Dual Data Rate (DDR) Input Register
-- UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

IDDRE1_inst : IDDRE1
generic map (
    DDR_CLK_EDGE => "OPPOSITE_EDGE", -- IDDRE1 mode (OPPOSITE_EDGE, SAME_EDGE, SAME_EDGE_PIPELINED)
    IS_CB_INVERTED => '0',             -- Optional inversion for CB
    IS_C_INVERTED => '0'              -- Optional inversion for C
)
port map (
    Q1 => Q1, -- 1-bit output: Registered parallel output 1
    Q2 => Q2, -- 1-bit output: Registered parallel output 2
    C => C,   -- 1-bit input: High-speed clock
    CB => CB, -- 1-bit input: Inversion of High-speed clock C
    D => D,   -- 1-bit input: Serial Data Input
    R => R    -- 1-bit input: Active High Async Reset
);

-- End of IDDRE1_inst instantiation
    
```

## Verilog Instantiation Template

```
// IDDRE1: Dedicated Dual Data Rate (DDR) Input Register
//      UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

IDDRE1 #(
    .DDR_CLK_EDGE("OPPOSITE_EDGE"), // IDDRE1 mode (OPPOSITE_EDGE, SAME_EDGE, SAME_EDGE_PIPELINED)
    .IS_CB_INVERTED(1'b0),          // Optional inversion for CB
    .IS_C_INVERTED(1'b0)           // Optional inversion for C
)
IDDRE1_inst (
    .Q1(Q1), // 1-bit output: Registered parallel output 1
    .Q2(Q2), // 1-bit output: Registered parallel output 2
    .C(C),   // 1-bit input: High-speed clock
    .CB(CB), // 1-bit input: Inversion of High-speed clock C
    .D(D),   // 1-bit input: Serial Data Input
    .R(R)    // 1-bit input: Active High Async Reset
);

// End of IDDRE1_inst instantiation
```

## For More Information

- See the *UltraScale Architecture SelectIO Resources User Guide* ([UG571](#)).
- See the [UltraScale Programmable Devices Documentation](#).

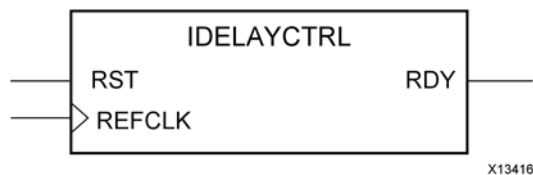
## IDELAYCTRL

Primitive: IDELAYE3/ODELAYE3 Tap Delay Value Control

PRIMITIVE\_GROUP: I/O

PRIMITIVE\_SUBGROUP: DELAY

Families: UltraScale, UltraScale+



### Introduction

At least one of these design elements must be instantiated when using IDELAYE3 or ODELAYE3. The IDELAYCTRL module provides a reference clock input that allows internal circuitry to define precise delay tap values independent of PVT (process, voltage, and temperature) for the IDELAYE3 and ODELAYE3 components.

### Port Descriptions

Port	Direction	Width	Function
RDY	Output	1	The ready (RDY) signal indicates when IDELAYE3 and ODELAYE3 modules in the specific region are calibrated. The RDY signal is deasserted if REFCLK is held High or Low for one clock more. If RDY is deasserted Low, the IDELAYCTRL module must be reset. If not needed, RDY to be unconnected/ignored.
REFCLK	Input	1	Time reference to IDELAYCTRL to calibrate all IDELAYE3 and ODELAYE3 modules in the same region. REFCLK can be supplied directly from a user-supplied source or the MMCME3/PLLE3 and must be routed on a global clock buffer.
RST	Input	1	Active-High reset. Asynchronous assertion, synchronous deassertion to REFCLK. To ensure proper IDELAYE3 and ODELAYE3 operation, IDELAYCTRL must be reset after configuration and the REFCLK signal is stable.

### Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	Yes



## VHDL 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: IDELAYE3/ODELAYE3 Tap Delay Value Control
--           UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

IDELAYCTRL_inst : IDELAYCTRL
generic map (
  SIM_DEVICE => "7SERIES" -- Set the device version (7SERIES, ULTRASCALE)
)
port map (
  RDY => RDY,           -- 1-bit output: Ready output
  REFCLK => REFCLK,     -- 1-bit input: Reference clock input
  RST => RST            -- 1-bit input: Active high reset input. Asynchronous assert, synchronous deassert to
                      -- REFCLK.
);

-- End of IDELAYCTRL_inst instantiation
  
```

## Verilog Instantiation Template

```

// IDELAYCTRL: IDELAYE3/ODELAYE3 Tap Delay Value Control
//           UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

IDELAYCTRL #(
  .SIM_DEVICE("7SERIES") // Set the device version (7SERIES, ULTRASCALE)
)
IDELAYCTRL_inst (
  .RDY(RDY),           // 1-bit output: Ready output
  .REFCLK(REFCLK),    // 1-bit input: Reference clock input
  .RST(RST)           // 1-bit input: Active high reset input. Asynchronous assert, synchronous deassert to
                      // REFCLK.
);

// End of IDELAYCTRL_inst instantiation
  
```

## For More Information

- See the *UltraScale Architecture SelectIO Resources User Guide* ([UG571](#)).
- See the [UltraScale Programmable Devices Documentation](#).

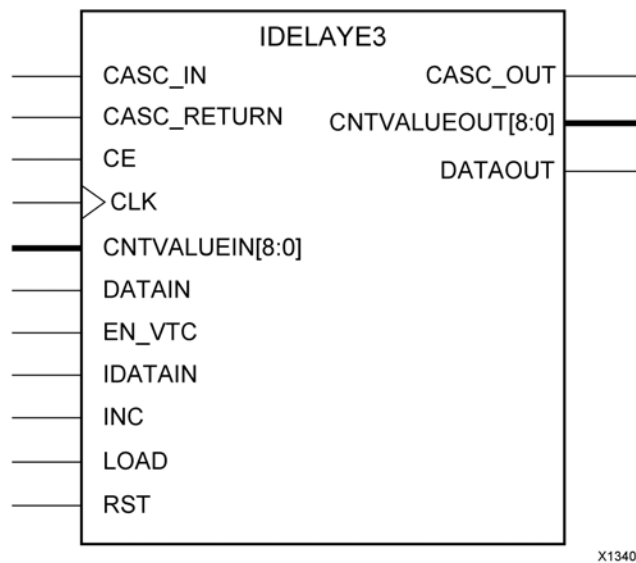
# IDELAYE3

Primitive: Input Fixed or Variable Delay Element

PRIMITIVE\_GROUP: I/O

PRIMITIVE\_SUBGROUP: DELAY

Families: UltraScale, UltraScale+



## Introduction

In component mode, I/O blocks contain a programmable delay element called IDELAYE3. The IDELAYE3 can be connected to an input register/ISERDESE3 or driven directly into device logic. The IDELAYE3 is a 512-tap delay element with a calibrated tap resolution. Refer to the device Data Sheet for delay values. The IDELAYE3 allows incoming signals to be delayed on an individual basis.

## Port Descriptions

Port	Direction	Width	Function
CASC_IN	Input	1	Cascade delay input from slave ODELAY CASCADE_OUT
CASC_OUT	Output	1	Cascade delay output to ODELAY input cascade
CASC_RETURN	Input	1	Cascade delay returning from slave ODELAY DATAOUT
CE	Input	1	Active high enable increment/decrement function
CLK	Input	1	Clock Input
CNTVALUEIN<8:0>	Input	9	Counter value from device logic for dynamically loadable tap value input
CNTVALUEOUT<8:0>	Output	9	The CNTVALUEOUT pins are used for reporting the dynamically switching value of the delay element. CNTVALUEOUT is only available when IDELAYE3 is in "VARIABLE" or "VAR_LOAD" mode.

Port	Direction	Width	Function
DATAIN	Input	1	The DATAIN input is directly driven by the device logic providing a logic accessible delay line. The data is driven back into the device logic through the DATAOUT port with a delay set by the DELAY_VALUE.
DATAOUT	Output	1	Delayed data output from one of two data input ports (IDATAIN or DATAIN).
EN_VTC	Input	1	Keep delay constant over VT
IDATAIN	Input	1	Data input for IDELAY from the IBUF
INC	Input	1	Increment / Decrement tap delay input
LOAD	Input	1	Loads the IDELAYE3 primitive to the pre-programmed value in VARIABLE mode. In VAR_LOAD mode, it load the value of CNTVALUEIN.
RST	Input	1	Asynchronous Reset to the DELAY_VALUE, active level based on IS_RST_INVERTED

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	Yes

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
CASCADE	STRING	"NONE", "MASTER", "SLAVE_END", "SLAVE_MIDDLE"	"NONE"	<p>Sets the location of the IDELAYE3 when it is used in a cascaded configuration</p> <ul style="list-style-type: none"> <li>"NONE" : Delay line is not cascaded</li> <li>"MASTER" : Delay line is cascaded with another delay line</li> <li>"SLAVE_MIDDLE" : Delay line is cascaded from adjacent delay line and also cascades to another delay line</li> <li>"SLAVE_END" : Delay line is the last cascaded delay line</li> </ul>

Attribute	Type	Allowed Values	Default	Description
DELAY_FORMAT	STRING	"TIME", "COUNT"	"TIME"	Sets the units of DELAY_VALUE of the IDELAYE3. Use TIME when DELAY_TYPE is FIXED. Use COUNT when DELAY_TYPE is VARIABLE or VAR_LOAD. <ul style="list-style-type: none"> <li>"TIME" : IDELAYE3 DELAY_VALUE is specified in ps</li> <li>"COUNT" : IDELAYE3 DELAY_VALUE is specified in taps</li> </ul>
DELAY_SRC	STRING	"IDATAIN", "DATAIN"	"IDATAIN"	Select the delay source input to the IDELAYE3 <ul style="list-style-type: none"> <li>"DATAIN" : IDELAYE3 chain input is DATAIN</li> <li>"IDATAIN": IDELAYE3 chain input is IDATAIN</li> </ul>
DELAY_TYPE	STRING	"FIXED", "VAR_LOAD", "VARIABLE"	"FIXED"	Sets the type of tap delay line. <ul style="list-style-type: none"> <li>"FIXED" - Sets a static delay value</li> <li>"VARIABLE" - Dynamically adjust (increment/decrement) delay value</li> <li>"VAR_LOAD" - Dynamically loads tap values</li> </ul>
DELAY_VALUE	DECIMAL	0 to 1250	0	Specifies the fixed delay in ps in "FIXED" mode or the initial starting number of taps in "VARIABLE" mode or "VAR_LOAD" mode (input path).
IS_CLK_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether the CLK pin is active-high or active-low.
IS_RST_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether the RST pin is active-high or active-low.
REFCLK_FREQUENCY	1 significant digit FLOAT	200.0 to 2667.0	300.0	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 performance.
SIM_DEVICE	STRING	"ULTRASCALE", "ULTRASCALE_PLUS_ES1"	"ULTRASCALE"	Set the device version

Attribute	Type	Allowed Values	Default	Description
UPDATE_MODE	STRING	"ASYNC", "MANUAL", "SYNC"	"ASYNC"	Determines when updates to the delay will take effect <ul style="list-style-type: none"> <li>"ASYNC": Updates are increments or decrements to the delay value independent of the data being received</li> <li>"SYNC": Updates require that DATAIN (or IDATAIN) transitions to synchronously update the delay with the DATAIN edges</li> <li>"MANUAL": Updates take effect when both LD and CE are asserted after the LD and CNTVALUEIN signals are used to load the new CNTVALUE</li> </ul>

## VHDL Instantiation Template

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- IDELAYE3: Input Fixed or Variable Delay Element
--           UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

IDELAYE3_inst : IDELAYE3
generic map (
    CASCADE => "NONE",           -- Cascade setting (MASTER, NONE, SLAVE_END, SLAVE_MIDDLE)
    DELAY_FORMAT => "TIME",      -- Units of the DELAY_VALUE (COUNT, TIME)
    DELAY_SRC => "IDATAIN",      -- Delay input (DATAIN, IDATAIN)
    DELAY_TYPE => "FIXED",      -- Set the type of tap delay line (FIXED, VARIABLE, VAR_LOAD)
    DELAY_VALUE => 0,           -- Input delay value setting
    IS_CLK_INVERTED => '0',     -- Optional inversion for CLK
    IS_RST_INVERTED => '0',     -- Optional inversion for RST
    REFCLK_FREQUENCY => 300.0,  -- IDELAYCTRL clock input frequency in MHz (200.0-2667.0)
    SIM_DEVICE => "ULTRASCALE", -- Set the device version (ULTRASCALE, ULTRASCALE_PLUS_ES1)
    UPDATE_MODE => "ASYNC"      -- Determines when updates to the delay will take effect (ASYNC, MANUAL,
                                -- SYNC)
)
port map (
    CASC_OUT => CASC_OUT,       -- 1-bit output: Cascade delay output to ODELAY input cascade
    CNTVALUEOUT => CNTVALUEOUT, -- 9-bit output: Counter value output
    DATAOUT => DATAOUT,      -- 1-bit output: Delayed data output
    CASC_IN => CASC_IN,        -- 1-bit input: Cascade delay input from slave ODELAY CASCADE_OUT
    CASC_RETURN => CASC_RETURN, -- 1-bit input: Cascade delay returning from slave ODELAY DATAOUT
    CE => CE,                  -- 1-bit input: Active high enable increment/decrement input
    CLK => CLK,                 -- 1-bit input: Clock input
    CNTVALUEIN => CNTVALUEIN,   -- 9-bit input: Counter value input
    DATAIN => DATAIN,        -- 1-bit input: Data input from the logic
    EN_VTC => EN_VTC,          -- 1-bit input: Keep delay constant over VT
    IDATAIN => IDATAIN,        -- 1-bit input: Data input from the IOBUF
    INC => INC,                 -- 1-bit input: Increment / Decrement tap delay input
    LOAD => LOAD,              -- 1-bit input: Load DELAY_VALUE input
    RST => RST                  -- 1-bit input: Asynchronous Reset to the DELAY_VALUE
);

-- End of IDELAYE3_inst instantiation
    
```

## Verilog Instantiation Template

```
// IDELAYE3: Input Fixed or Variable Delay Element
// UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

IDELAYE3 #(
    .CASCADE("NONE"),           // Cascade setting (MASTER, NONE, SLAVE_END, SLAVE_MIDDLE)
    .DELAY_FORMAT("TIME"),     // Units of the DELAY_VALUE (COUNT, TIME)
    .DELAY_SRC("IDATAIN"),     // Delay input (DATAIN, IDATAIN)
    .DELAY_TYPE("FIXED"),      // Set the type of tap delay line (FIXED, VARIABLE, VAR_LOAD)
    .DELAY_VALUE(0),           // Input delay value setting
    .IS_CLK_INVERTED(1'b0),    // Optional inversion for CLK
    .IS_RST_INVERTED(1'b0),    // Optional inversion for RST
    .REFCLK_FREQUENCY(300.0),  // IDELAYCTRL clock input frequency in MHz (200.0-2667.0)
    .SIM_DEVICE("ULTRASCALE"), // Set the device version (ULTRASCALE, ULTRASCALE_PLUS_ES1)
    .UPDATE_MODE("ASYNCR")    // Determines when updates to the delay will take effect (ASYNCR, MANUAL, SYNC)
)
IDELAYE3_inst (
    .CASC_OUT(CASC_OUT),       // 1-bit output: Cascade delay output to ODELAY input cascade
    .CNTVALUEOUT(CNTVALUEOUT), // 9-bit output: Counter value output
    .DATAOUT(DATAOUT),         // 1-bit output: Delayed data output
    .CASC_IN(CASC_IN),         // 1-bit input: Cascade delay input from slave ODELAY CASCADE_OUT
    .CASC_RETURN(CASC_RETURN), // 1-bit input: Cascade delay returning from slave ODELAY DATAOUT
    .CE(CE),                   // 1-bit input: Active high enable increment/decrement input
    .CLK(CLK),                  // 1-bit input: Clock input
    .CNTVALUEIN(CNTVALUEIN),   // 9-bit input: Counter value input
    .DATAIN(DATAIN),           // 1-bit input: Data input from the logic
    .EN_VTC(EN_VTC),           // 1-bit input: Keep delay constant over VT
    .IDATAIN(IDATAIN),         // 1-bit input: Data input from the IOBUF
    .INC(INC),                  // 1-bit input: Increment / Decrement tap delay input
    .LOAD(LOAD),               // 1-bit input: Load DELAY_VALUE input
    .RST(RST)                  // 1-bit input: Asynchronous Reset to the DELAY_VALUE
);

// End of IDELAYE3_inst instantiation
```

## For More Information

- See the *UltraScale Architecture SelectIO Resources User Guide* ([UG571](#)).
- See the [UltraScale Programmable Devices Documentation](#).

# ILKN

Primitive: Interlaken MAC

PRIMITIVE\_GROUP: **ADVANCED**

PRIMITIVE\_SUBGROUP: INTERLAKEN

Families: UltraScale

## Introduction

The Interlaken protocol block provides a high-performance, low power implementation of the Interlaken protocol that provides a low risk, quick path for adopting Interlaken as a chip-to-chip interconnect protocol. This block is designed to be integrated with GTs and device clocking resources using fabric interconnect. See PG169: UltraScale Architecture Integrated IP Core for Interlaken v1.4 Product Guide for further details on this component

## Design Entry Method

Instantiation	No
Inference	No
IP and IP Integrator Catalog	Recommended

## For More Information

- See the *Integrated Interlaken up to 150G v1.7 Product Guide* ([PG169](#)).
- See the [UltraScale Programmable Devices Documentation](#).

# ILKNE4

Primitive: Interlaken MAC

PRIMITIVE\_GROUP: **ADVANCED**

PRIMITIVE\_SUBGROUP: INTERLAKEN

Families: UltraScale+

## Introduction

The Interlaken protocol block provides a high-performance, low power implementation of the Interlaken protocol that provides a low risk, quick path for adopting Interlaken as a chip-to-chip interconnect protocol. This block is designed to be integrated with GTs and device clocking resources using fabric interconnect. See the UltraScale+ Architecture Integrated IP Core for Interlaken Product Guide for further details on this component

## Design Entry Method

Instantiation	No
Inference	No
IP and IP Integrator Catalog	Recommended

## For More Information

- See the *Integrated Interlaken up to 150G v1.7 Product Guide* ([PG169](#)).
- See the [UltraScale Programmable Devices Documentation](#).



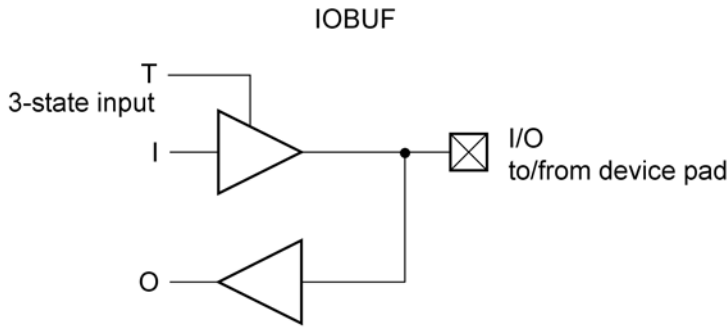
# IOBUF

## Primitive: Input/Output Buffer

PRIMITIVE\_GROUP: I/O

PRIMITIVE\_SUBGROUP: BIDIR\_BUFFER

Families: UltraScale, UltraScale+



X10663

## Introduction

The IOBUF primitive is needed when bidirectional signals require both an input buffer and a 3-state output buffer with an active-High 3-state T pin. The IOBUF is a generic IOBUF. A logic-High on the T pin disables the output buffer. When the output buffer is 3-stated (T = High), the input buffer and any on-die receiver termination (uncalibrated or DCI) are ON. When the output buffer is not 3-stated (T = Low), any on-die receiver termination (uncalibrated or DCI) is disabled.

I/O attributes that do not impact the logic function of the component such as IOSTANDARD, DRIVE and SLEW should be supplied to the top-level port via an appropriate property. For details on applying such properties to the associated port, see UG912: Vivado Design Suite Properties Reference Guide.

## 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
I	Input	1	Input of OBUF. Connect to the logic driving the output port.
IO	Inout	1	Bidirectional port to be connected directly to top-level inout port.
O	Output	1	Output path of the buffer.
T	Input	1	3-state enable input signifying whether the buffer acts as an input or output.

## Design Entry Method

Instantiation	Yes
Inference	Recommended
IP and IP Integrator Catalog	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;

-- IOBUF: Input/Output Buffer
--   UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

IOBUF_inst : IOBUF
port map (
  O => O,    -- 1-bit output: Buffer output
  I => I,    -- 1-bit input: Buffer input
  IO => IO,  -- 1-bit inout: Buffer inout (connect directly to top-level port)
  T => T     -- 1-bit input: 3-state enable input
);

-- End of IOBUF_inst instantiation
    
```

## Verilog Instantiation Template

```

// IOBUF: Input/Output Buffer
//   UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

IOBUF IOBUF_inst (
  .O(O),    // 1-bit output: Buffer output
  .I(I),    // 1-bit input: Buffer input
  .IO(IO),  // 1-bit inout: Buffer inout (connect directly to top-level port)
  .T(T)     // 1-bit input: 3-state enable input
);

// End of IOBUF_inst instantiation
    
```

## For More Information

- See the *UltraScale Architecture SelectIO Resources User Guide* ([UG571](#)).
- See the [UltraScale Programmable Devices Documentation](#).

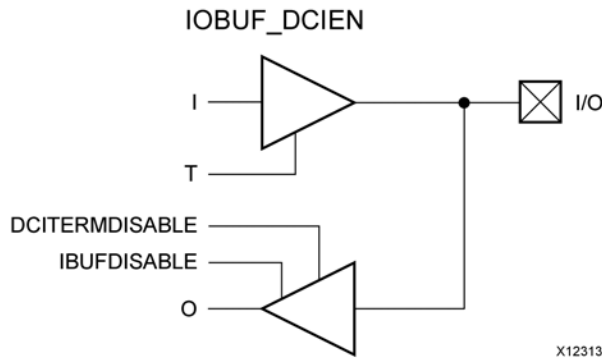
## IOBUF\_DCIEN

Primitive: Input/Output Buffer DCI Enable

PRIMITIVE\_GROUP: I/O

PRIMITIVE\_SUBGROUP: BIDIR\_BUFFER

Families: UltraScale, UltraScale+



### Introduction

The IOBUF\_DCIEN primitive is available in the HP I/O banks. It has an IBUFDISABLE port that can be used to disable the input buffer during periods that the buffer is not being used. The IOBUF\_DCIEN primitive also has a DCITERMDISABLE port that can be used to manually disable the optional on-die receiver termination features (uncalibrated and DCI). See UG571: UltraScale Architecture SelectIO Resources, "DCI Only available in the HP I/O Banks" and "Uncalibrated Input Termination in I/O Banks" sections for more details.

The IOBUF\_DCIEN primitive can disable the input buffer and force the O output to the internal logic to a logic-Low when the IBUFDISABLE signal is asserted High and output buffer is 3-stated (T = High). If the I/O is using any on-die receiver termination features (uncalibrated and DCI), this primitive disables the termination legs whenever the DCITERMDISABLE signal is asserted High and the output buffer is 3-stated (T = High). When the output buffer is 3-stated (T = High), the input buffer and any on-die receiver termination (uncalibrated or DCI) are controlled by IBUFDISABLE and DCITERMDISABLE, respectively. The USE\_IBUFDISABLE attribute must be set to TRUE and SIM\_DEVICE to ULTRASCALE for this primitive to have the expected behavior that is specific to the UltraScale architecture. When the output buffer is not 3-stated (T = Low), the input buffer and any on-die receiver termination (uncalibrated or DCI) are disabled and the O output (to the internal logic) is forced to a logic-Low. These features can be combined to reduce power whenever the input is idle for a period of time.

I/O attributes that do not impact the logic function of the component, such as IOSTANDARD, DRIVE, and SLEW, should be supplied to the top-level port via an appropriate property. For details on applying such properties to the associated port, see UG912: Vivado Design Suite Properties Reference Guide.

## Port Descriptions

Port	Direction	Width	Function
DCITERMDISABLE	Input	1	Control to enable/disable DCI termination. This is generally used to reduce power in long periods of an idle state.
I	Input	1	Input of OBUF. Connect to the logic driving the output port.
IBUFDISABLE	Input	1	Disables input path through the buffer and forces to a logic low. This feature is generally used to reduce power at times when the I/O is idle for a period of time.
IO	Inout	1	Bidirectional port to be connected directly to top-level inout port.
O	Output	1	Output path of the buffer.
T	Input	1	3-state enable input signifying whether the buffer acts as an input or output.

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
SIM_DEVICE	STRING	"ULTRASCALE", "7SERIES"	"7SERIES"	This attribute must be set to "ULTRASCALE" in order to exhibit the proper simulation behavior.
USE_IBUFDISABLE	STRING	"TRUE", "FALSE"	"TRUE"	This attribute must be unspecified or set to "TRUE" if specified.

## VHDL 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_DCIEN: Input/Output Buffer DCI Enable
--           UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

IOBUF_DCIEN_inst : IOBUF_DCIEN
generic map (
    SIM_DEVICE => "ULTRASCALE" -- Must be set to "ULTRASCALE"
)
port map (
    O => O, -- 1-bit output: Buffer output
    DCITERMDISABLE => DCITERMDISABLE, -- 1-bit input: DCI Termination Disable
    I => I, -- 1-bit input: Buffer input
    IBUFDISABLE => IBUFDISABLE, -- 1-bit input: Buffer disable input, high=disable
    IO => IO, -- 1-bit inout: Buffer inout (connect directly to top-level port)
    T => T -- 1-bit input: 3-state enable input
    
```

```
);  
-- End of IOBUF_DCIEN_inst instantiation
```

## Verilog Instantiation Template

```
// IOBUF_DCIEN: Input/Output Buffer DCI Enable  
// UltraScale  
// Xilinx HDL Libraries Guide, version 2016.1  
  
IOBUF_DCIEN #(  
    .SIM_DEVICE("ULTRASCALE") // Must be set to "ULTRASCALE"  
)  
IOBUF_DCIEN_inst (  
    .O(O), // 1-bit output: Buffer output  
    .DCITERMDISABLE(DCITERMDISABLE), // 1-bit input: DCI Termination Disable  
    .I(I), // 1-bit input: Buffer input  
    .IBUFDISABLE(IBUFDISABLE), // 1-bit input: Buffer disable input, high=disable  
    .IO(IO), // 1-bit inout: Buffer inout (connect directly to top-level port)  
    .T(T) // 1-bit input: 3-state enable input  
)  
);  
  
// End of IOBUF_DCIEN_inst instantiation
```

## For More Information

- See the *UltraScale Architecture SelectIO Resources User Guide* ([UG571](#)).
- See the [UltraScale Programmable Devices Documentation](#).

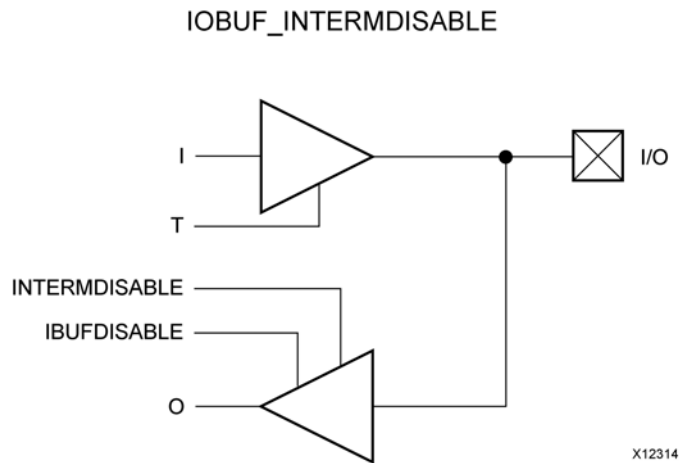
## IOBUF\_INTERMDISABLE

Primitive: Bidirectional Buffer with Input Path Disable and On-die Input Termination Disable

PRIMITIVE\_GROUP: I/O

PRIMITIVE\_SUBGROUP: BIDIR\_BUFFER

Families: UltraScale, UltraScale+



### Introduction

The IOBUF\_INTERMDISABLE primitive is available in the HR I/O banks. It has an IBUFDISABLE port that can be used to disable the input buffer during periods that the buffer is not being used. The IOBUF\_INTERMDISABLE primitive also has an INTERMDISABLE port that can be used to manually disable the optional on-die receiver termination feature. See UG571: UltraScale Architecture SelectIO Resources, "Uncalibrated Input Termination in I/O Banks" for more details.

The IOBUF\_INTERMDISABLE primitive can disable the input buffer and force the O output to the internal logic to a logic-Low when the IBUFDISABLE signal is asserted High and the output buffer is 3-stated (T = High). If the I/O is using the on-die receiver termination feature (uncalibrated), this primitive disables the termination legs whenever the INTERMDISABLE signal is asserted High and the output buffer is 3-stated (T = High). When the output buffer is 3-stated (T = High), the input buffer and any on-die receiver termination are controlled by IBUFDISABLE and INTERMDISABLE, respectively. The USE\_IBUFDISABLE attribute must be set to TRUE and the SIM\_DEVICE set to ULTRASCALE for this primitive to have the expected behavior that is specific to the UltraScale architecture. When the output buffer is not 3-stated (T = Low), the input buffer and any on-die receiver termination are disabled and the O output (to the internal logic) is forced to a logic-Low. These features can be combined to reduce power whenever the input is idle for a period of time.

I/O attributes that do not impact the logic function of the component, such as IOSTANDARD and IBUF\_LOW\_PWR, should be supplied to the top-level port via an appropriate property. For details on applying such properties to the associated port, see UG912: Vivado Design Suite Properties Reference Guide.

## Port Descriptions

Port	Direction	Width	Function
I	Input	1	Buffer input representing the output path from the device.
IBUFDISABLE	Input	1	Disables input path through the buffer and forces to a logic low. This feature is generally used to reduce power at times when the I/O is idle for a period of time.
INTERMDISABLE	Input	1	Control to enable/disable on-chip input termination. This is generally used to reduce power in long periods of an idle state.
IO	Inout	1	Bidirectional I/O port connection. Connect directly to top-level port in the design.
O	Output	1	Buffer output representing the input path to the device.
T	Input	1	3-state enable input signifying whether the buffer acts as an input or output.

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
SIM_DEVICE	STRING	"ULTRASCALE", "7SERIES"	"7SERIES"	This attribute must be set to "ULTRASCALE" in order to exhibit the proper simulation behavior.
USE_IBUFDISABLE	STRING	"TRUE", "FALSE"	"TRUE"	This attribute must be unspecified or set to "TRUE" if specified.

## VHDL 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_INTERMDISABLE: Bidirectional Buffer with Input Path Disable and On-die Input Termination Disable
--                               UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

IOBUF_INTERMDISABLE_inst : IOBUF_INTERMDISABLE
generic map (
    SIM_DEVICE => "ULTRASCALE" -- Must be set to "ULTRASCALE"
)
port map (
    O => O,           -- 1-bit output: Buffer output
    I => I,           -- 1-bit input: Buffer input
    IBUFDISABLE => IBUFDISABLE, -- 1-bit input: Buffer disable input, high=disable
    INTERMDISABLE => INTERMDISABLE, -- 1-bit input: Input Termination Disable
    IO => IO,         -- 1-bit inout: Buffer inout (connect directly to top-level port)
    T => T            -- 1-bit input: 3-state enable input
    
```



```
);  
-- End of IOBUF_INTERMDISABLE_inst instantiation
```

## Verilog Instantiation Template

```
// IOBUF_INTERMDISABLE: Bidirectional Buffer with Input Path Disable and On-die Input Termination Disable  
// UltraScale  
// Xilinx HDL Libraries Guide, version 2016.1  
  
IOBUF_INTERMDISABLE #(  
    .SIM_DEVICE("ULTRASCALE") // Must be set to "ULTRASCALE"  
)  
IOBUF_INTERMDISABLE_inst (  
    .O(O), // 1-bit output: Buffer output  
    .I(I), // 1-bit input: Buffer input  
    .IBUFDISABLE(IBUFDISABLE), // 1-bit input: Buffer disable input, high=disable  
    .INTERMDISABLE(INTERMDISABLE), // 1-bit input: Input Termination Disable  
    .IO(IO), // 1-bit inout: Buffer inout (connect directly to top-level port)  
    .T(T) // 1-bit input: 3-state enable input  
)  
);  
  
// End of IOBUF_INTERMDISABLE_inst instantiation
```

## For More Information

- See the *UltraScale Architecture SelectIO Resources User Guide* ([UG571](#)).
- See the [UltraScale Programmable Devices Documentation](#).

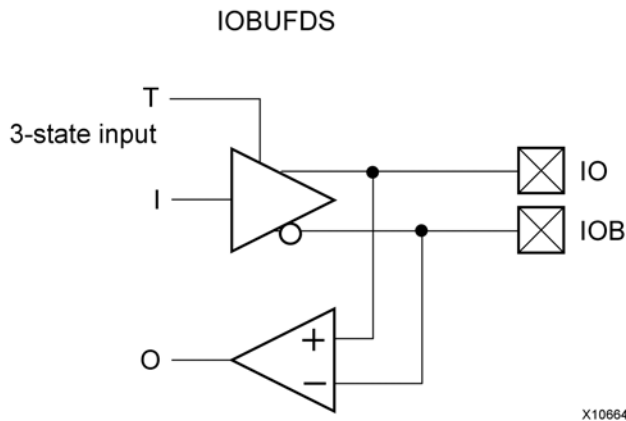
# IOBUFDS

Primitive: Differential Input/Output Buffer

PRIMITIVE\_GROUP: I/O

PRIMITIVE\_SUBGROUP: BIDIR\_BUFFER

Families: UltraScale, UltraScale+



## Introduction

The IOBUFDS is a differential input/output buffer primitive. A logic-High on the T pin disables the output buffer. When the output buffer is 3-stated (T = High), the input buffer and any on-die receiver termination (uncalibrated or DCI) are ON. When the output buffer is not 3-stated (T = Low), any on-die receiver termination (uncalibrated or DCI) is disabled.

I/O attributes that do not impact the logic function of the component, such as IOSTANDARD, DRIVE, and SLEW, should be supplied to the top-level port via an appropriate property. For details on applying such properties to the associated port, see UG912: Vivado Design Suite Properties Reference Guide.

## Logic Table

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

## Port Descriptions

Port	Direction	Width	Function
I	Input	1	Input of OBUF. Connect to the logic driving the output port.
IO	Inout	1	Bidirectional diff_p port to be connected directly to top-level inout port.
IOB	Inout	1	Bidirectional diff_n port to be connected directly to top-level inout port.
O	Output	1	Output path of the buffer.
T	Input	1	3-state enable input signifying whether the buffer acts as an input or output.

## Design Entry Method

Instantiation	Recommended
Inference	No
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
DQS_BIAS	STRING	"FALSE", "TRUE"	"FALSE"	Provides pull-up/pull-down feature required for some DQS memory interface pins.

## VHDL 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 Input/Output Buffer
-- UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

IOBUFDS_inst : IOBUFDS
generic map (
    DQS_BIAS => "FALSE" -- (FALSE, TRUE)
)
port map (
    O => O,      -- 1-bit output: Buffer output
    I => I,      -- 1-bit input: Buffer input
    IO => IO,    -- 1-bit inout: Diff_p inout (connect directly to top-level port)
    IOB => IOB, -- 1-bit inout: Diff_n inout (connect directly to top-level port)
    T => T      -- 1-bit input: 3-state enable input
);

-- End of IOBUFDS_inst instantiation
    
```

## Verilog Instantiation Template

```
// IOBUFDS: Differential Input/Output Buffer
// UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

IOBUFDS #(
    .DQS_BIAS("FALSE") // (FALSE, TRUE)
)
IOBUFDS_inst (
    .O(O), // 1-bit output: Buffer output
    .I(I), // 1-bit input: Buffer input
    .IO(IO), // 1-bit inout: Diff_p inout (connect directly to top-level port)
    .IOB(IOB), // 1-bit inout: Diff_n inout (connect directly to top-level port)
    .T(T) // 1-bit input: 3-state enable input
);

// End of IOBUFDS_inst instantiation
```

## For More Information

- See the *UltraScale Architecture SelectIO Resources User Guide* ([UG571](#)).
- See the [UltraScale Programmable Devices Documentation](#).

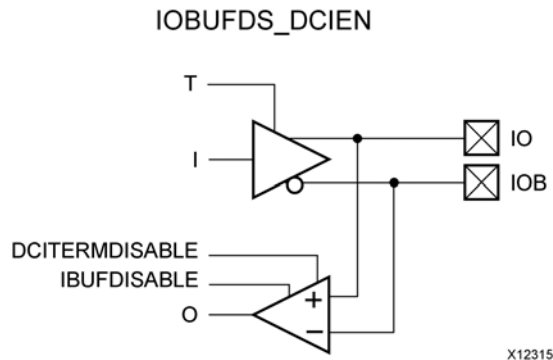
## IOBUFDS\_DCIEN

**Primitive: Differential Bidirectional Buffer With Input Buffer Disable and On-die Input Termination Disable**

PRIMITIVE\_GROUP: I/O

PRIMITIVE\_SUBGROUP: BIDIR\_BUFFER

Families: UltraScale, UltraScale+



### Introduction

The IOBUFDS\_DCIEN primitive is available in the HP I/O banks. It has an IBUFDISABLE port that can be used to disable the input buffer during periods that the buffer is not being used. The USE\_IBUFDISABLE attribute must be set to TRUE and SIM\_DEVICE to ULTRASCALE for this primitive to have the expected behavior that is specific to the UltraScale architecture. The IOBUFDS\_DCIEN primitive also has a DCITERMDISABLE port that can be used to manually disable the optional on-die receiver termination features (uncalibrated or DCI). See UG571: UltraScale Architecture SelectIO Resources, "DCI Only available in the HP I/O Banks" and "Uncalibrated Input Termination in I/O Banks" sections for more details.

The IOBUFDS\_DCIEN primitive can disable the input buffer and force the O output to the internal logic to a logic-Low when the IBUFDISABLE signal is asserted High and the output buffer is 3-stated (T = High). If the I/O is using an on-die receiver termination feature (uncalibrated or DCI), this primitive disables the termination legs whenever the DCITERMDISABLE signal is asserted High and the output buffer is 3-stated (T = High).

When the output buffer is 3-stated (T = High), the input buffer and any on-die receiver termination (uncalibrated or DCI) are controlled by IBUFDISABLE and DCITERMDISABLE, respectively. When the output buffer is not 3-stated (T = Low), the input buffer and any on-die receiver termination (uncalibrated or DCI) are disabled and force the O output (to the internal logic) to a logic-Low. These features can be combined to reduce power whenever the input is idle for a period of time.

I/O attributes that do not impact the logic function of the component, such as IOSTANDARD, DRIVE, and SLEW, should be supplied to the top-level port via an appropriate property. For details on applying such properties to the associated port, see UG912: Vivado Design Suite Properties Reference Guide. Attributes that impact the functionality, such as DQS\_BIAS, must be supplied to the component via a generic\_map (VHDL) or parameter (Verilog) in order to have the correct simulation behavior.

## Port Descriptions

Port	Direction	Width	Function
DCITERMDISABLE	Input	1	Control to enable/disable DCI termination. This is generally used to reduce power in long periods of an idle state.
I	Input	1	Input of OBUF. Connect to the logic driving the output port.
IBUFDISABLE	Input	1	Disables input path through the buffer and forces to a logic low. This feature is generally used to reduce power at times when the I/O is idle for a period of time.
IO	Inout	1	Bidirectional diff_p port to be connected directly to top-level inout port.
IOB	Inout	1	Bidirectional diff_n port to be connected directly to top-level inout port.
O	Output	1	Output path of the buffer.
T	Input	1	3-state enable input signifying whether the buffer acts as an input or output.

## Design Entry Method

Instantiation	Recommended
Inference	No
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
DQS_BIAS	STRING	"FALSE", "TRUE"	"FALSE"	Provides pull-up/pull-down feature required for some DQS memory interface pins.
SIM_DEVICE	STRING	"ULTRASCALE", "7SERIES"	"7SERIES"	This attribute must be set to "ULTRASCALE" in order to exhibit the proper simulation behavior.
USE_IBUFDISABLE	STRING	"TRUE", "FALSE"	"TRUE"	This attribute must be unspecified or set to "TRUE" if specified.

## VHDL 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_DCIEN: Differential Bidirectional Buffer With Input Buffer Disable and On-die Input Termination Disable
-- UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

IOBUFDS_DCIEN_inst : IOBUFDS_DCIEN
generic map (
    DQS_BIAS => "FALSE",      -- (FALSE, TRUE)
    SIM_DEVICE => "ULTRASCALE" -- Must be set to "ULTRASCALE"

```

```

)
port map (
  O => O,           -- 1-bit output: Buffer output
  DCITERMDISABLE => DCITERMDISABLE, -- 1-bit input: DCI Termination Disable
  I => I,           -- 1-bit input: Buffer input
  IBUFDISABLE => IBUFDISABLE,       -- 1-bit input: Buffer disable input, high=disable
  IO => IO,         -- 1-bit inout: Diff_p inout (connect directly to top-level port)
  IOB => IOB,       -- 1-bit inout: Diff_n inout (connect directly to top-level port)
  T => T           -- 1-bit input: 3-state enable input
);

-- End of IOBUFDS_DCIEN_inst instantiation

```

## Verilog Instantiation Template

```

// IOBUFDS_DCIEN: Differential Bidirectional Buffer With Input Buffer Disable and On-die Input Termination Disable
//                               UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

IOBUFDS_DCIEN #(
  .DQS_BIAS("FALSE"),           // (FALSE, TRUE)
  .SIM_DEVICE("ULTRASCALE")    // Must be set to "ULTRASCALE"
)
IOBUFDS_DCIEN_inst (
  .O(O),           // 1-bit output: Buffer output
  .DCITERMDISABLE(DCITERMDISABLE), // 1-bit input: DCI Termination Disable
  .I(I),           // 1-bit input: Buffer input
  .IBUFDISABLE(IBUFDISABLE),       // 1-bit input: Buffer disable input, high=disable
  .IO(IO),         // 1-bit inout: Diff_p inout (connect directly to top-level port)
  .IOB(IOB),       // 1-bit inout: Diff_n inout (connect directly to top-level port)
  .T(T)           // 1-bit input: 3-state enable input
);

// End of IOBUFDS_DCIEN_inst instantiation

```

## For More Information

- See the *UltraScale Architecture SelectIO Resources User Guide* ([UG571](#)).
- See the [UltraScale Programmable Devices Documentation](#).

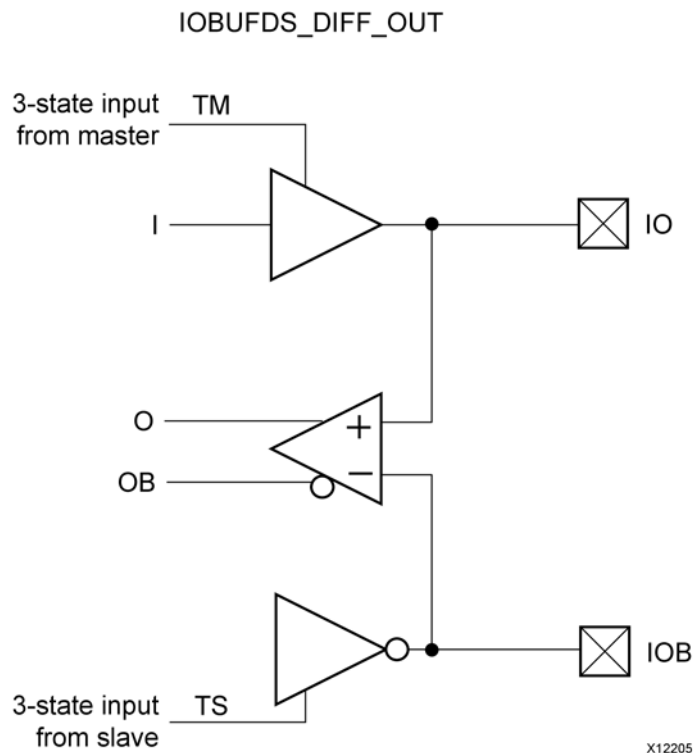
## IOBUFDS\_DIFF\_OUT

Primitive: Differential Input/Output Buffer Primitive With Complementary Outputs for the Input Buffer

PRIMITIVE\_GROUP: I/O

PRIMITIVE\_SUBGROUP: BIDIR\_BUFFER

Families: UltraScale, UltraScale+



### Introduction

The IOBUFDS\_DIFF\_OUT is a differential input/output buffer primitive with complementary outputs (O and OB). A logic-High on the T pin disables the output buffer. When the output buffer is 3-stated (T = High), the input buffer and any on-die receiver termination (uncalibrated or DCI) are ON. When the output buffer is not 3-stated (T = Low), any on-die receiver termination (uncalibrated or DCI) is disabled. TM and TS must be connected to the same input from the interconnect logic for this primitive to have the expected behavior that is specific to the UltraScale architecture.

I/O attributes that do not impact the logic function of the component, such as IOSTANDARD, DIFF\_TERM, and IBUF\_LOW\_PWR, should be supplied to the top-level port via an appropriate property. For details on applying such properties to the associated port, see UG912: Vivado Design Suite Properties Reference Guide. Attributes that impact the functionality such as DQS\_BIAS must be supplied to the component via a generic\_map (VHDL) or parameter (Verilog) in order to have the correct simulation behavior.



## Port Descriptions

Port	Direction	Width	Function
I	Input	1	Input of OBUF. Connect to the logic driving the output port.
IO	Inout	1	Bidirectional diff_p port to be connected directly to top-level inout port.
IOB	Inout	1	Bidirectional diff_n port to be connected directly to top-level inout port.
O	Output	1	Output path of the buffer.
OB	Output	1	Output path of the buffer.
TM	Input	1	3-state enable input for the p-side or master side signifying whether the buffer acts as an input or output. This pin must be connected to the same signal as the TS input.
TS	Input	1	3-state enable input for the n-side or slave side signifying whether the buffer acts as an input or output. This pin must be connected to the same signal as the TM input.

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
DQS_BIAS	STRING	"FALSE", "TRUE"	"FALSE"	Provides pull-up/pull-down feature required for some DQS memory interface pins.

## VHDL 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_DIFF_OUT: Differential Input/Output Buffer Primitive With Complementary Outputs for the Input Buffer
-- UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

IOBUFDS_DIFF_OUT_inst : IOBUFDS_DIFF_OUT
generic map (
    DQS_BIAS => "FALSE" -- (FALSE, TRUE)
)
port map (
    O => O,      -- 1-bit output: Buffer diff_p output
    OB => OB,    -- 1-bit output: Buffer diff_n output
    I => I,      -- 1-bit input: Buffer input
    IO => IO,    -- 1-bit inout: Diff_p inout (connect directly to top-level port)
    IOB => IOB,  -- 1-bit inout: Diff_n inout (connect directly to top-level port)
    TM => TM,    -- 1-bit input: 3-state master enable input
    
```

```

    TS => TS    -- 1-bit input: 3-state slave enable input
  );
-- End of IOBUFDS_DIFF_OUT_inst instantiation

```

## Verilog Instantiation Template

```

// IOBUFDS_DIFF_OUT: Differential Input/Output Buffer Primitive With Complementary Outputs for the Input Buffer
//                               UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

IOBUFDS_DIFF_OUT #(
    .DQS_BIAS("FALSE") // (FALSE, TRUE)
)
IOBUFDS_DIFF_OUT_inst (
    .O(O), // 1-bit output: Buffer diff_p output
    .OB(OB), // 1-bit output: Buffer diff_n output
    .I(I), // 1-bit input: Buffer input
    .IO(IO), // 1-bit inout: Diff_p inout (connect directly to top-level port)
    .IOB(IOB), // 1-bit inout: Diff_n inout (connect directly to top-level port)
    .TM(TM), // 1-bit input: 3-state master enable input
    .TS(TS) // 1-bit input: 3-state slave enable input
);
// End of IOBUFDS_DIFF_OUT_inst instantiation

```

## For More Information

- See the *UltraScale Architecture SelectIO Resources User Guide* ([UG571](#)).
- See the [UltraScale Programmable Devices Documentation](#).

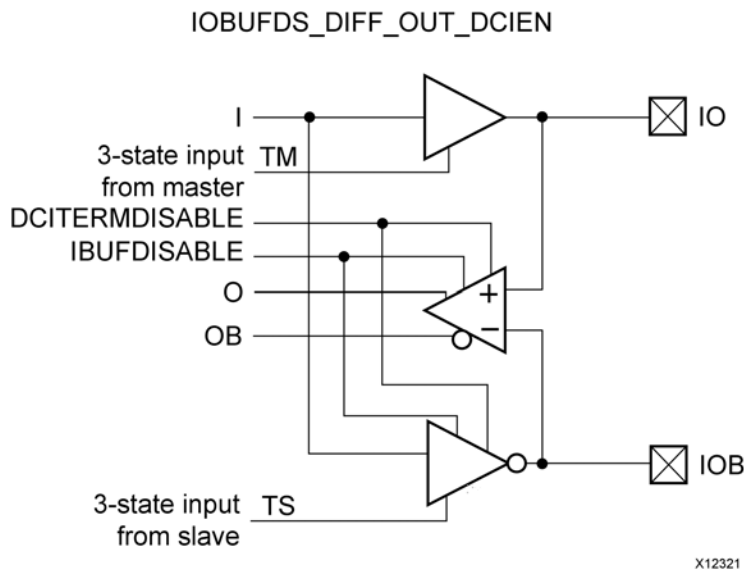
## IOBUFDS\_DIFF\_OUT\_DCIEEN

Primitive: Differential Bidirectional Buffer with Complementary Outputs, Input Path Disable, and On-die Input Termination Disable

PRIMITIVE\_GROUP: I/O

PRIMITIVE\_SUBGROUP: BIDIR\_BUFFER

Families: UltraScale, UltraScale+



### Introduction

The IOBUFDS\_DIFF\_OUT\_DCIEEN primitive is available in the HP I/O banks. It has complementary differential outputs, an IBUFDISABLE port, and a DCITERMDISABLE port that can be used to manually disable the optional DCI on-die receiver termination features (uncalibrated or DCI). See UG571: UltraScale Architecture SelectIO Resources, "DCI Only available in the HP I/O Banks" and "Uncalibrated Input Termination in I/O Banks" sections for more details. The USE\_IBUFDISABLE attribute must be set to TRUE and SIM\_DEVICE to ULTRASCALE for this primitive to have the expected behavior that is specific to the UltraScale architecture. TM and TS must be connected to the same input from the interconnect logic for this primitive to have the expected behavior that is specific to the UltraScale architecture.

If the I/O is using any on-die receiver termination features (uncalibrated or DCI), this primitive disables the termination legs whenever the DCITERMDISABLE signal is asserted High and the output buffer is 3-stated. When the output buffer is 3-stated (T = High), any on-die receiver termination (uncalibrated or DCI) is controlled by DCITERMDISABLE. When the output buffer is not 3-stated (T = Low), the input buffer and on-die receiver termination (uncalibrated or DCI) are disabled and the O output (to the internal logic) is forced to a logic-Low.

I/O attributes that do not impact the logic function of the component, such as IOSTANDARD, DIFF\_TERM, and IBUF\_LOW\_PWR, should be supplied to the top-level port via an appropriate property. For details on applying such properties to the associated port, see UG912: Vivado Design Suite Properties Reference Guide. Attributes that impact the functionality such as DQS\_BIAS must be supplied to the component via a generic\_map (VHDL) or parameter (Verilog) in order to have the correct simulation behavior.

## Port Descriptions

Port	Direction	Width	Function
DCITERMDISABLE	Input	1	Control to enable/disable DCI termination. This is generally used to reduce power in long periods of an idle state.
I	Input	1	Input of OBUF. Connect to the logic driving the output port.
IBUFDISABLE	Input	1	The IBUFDISABLE feature is not supported with this primitive in the UltraScale architecture. This port must be tied to logic '0'.
IO	Inout	1	Bidirectional diff_p port to be connected directly to top-level inout port.
IOB	Inout	1	Bidirectional diff_n port to be connected directly to top-level inout port.
O	Output	1	Output path of the buffer.
OB	Output	1	Output path of the buffer.
TM	Input	1	3-state enable input for the p-side or master side signifying whether the buffer acts as an input or output. This pin must be connected to the same signal as the TS input.
TS	Input	1	3-state enable input for the n-side or slave side signifying whether the buffer acts as an input or output. This pin must be connected to the same signal as the TM input.

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
DQS_BIAS	STRING	"FALSE", "TRUE"	"FALSE"	Provides pull-up/pull-down feature required for some DQS memory interface pins.
SIM_DEVICE	STRING	"ULTRASCALE", "7SERIES"	"7SERIES"	This attribute must be set to "ULTRASCALE" in order to exhibit the proper simulation behavior.
USE_IBUFDISABLE	STRING	"TRUE", "FALSE"	"TRUE"	This attribute must be unspecified or set to "TRUE" if specified.

## VHDL 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_DIFF_OUT_DCIEN: Differential Bidirectional Buffer with Complementary Outputs, Input Path Disable, and On-die Input Termination Disable
--
-- UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

IOBUFDS_DIFF_OUT_DCIEN_inst : IOBUFDS_DIFF_OUT_DCIEN
generic map (
    DQS_BIAS => "FALSE",          -- (FALSE, TRUE)
    SIM_DEVICE => "ULTRASCALE"   -- Must be set to "ULTRASCALE"
)
port map (
    O => O,                       -- 1-bit output: Buffer diff_p output
    OB => OB,                     -- 1-bit output: Buffer diff_n output
    DCITERMDISABLE => DCITERMDISABLE, -- 1-bit input: DCI Termination Disable
    I => I,                       -- 1-bit input: Buffer input
    IBUFDISABLE => IBUFDISABLE,   -- 1-bit input: Must be tied to a logic '0'
    IO => IO,                    -- 1-bit inout: Diff_p inout (connect directly to top-level port)
    IOB => IOB,                  -- 1-bit inout: Diff_n inout (connect directly to top-level port)
    TM => TM,                    -- 1-bit input: 3-state master enable input
    TS => TS                     -- 1-bit input: 3-state slave enable input
);

-- End of IOBUFDS_DIFF_OUT_DCIEN_inst instantiation
    
```

## Verilog Instantiation Template

```

// IOBUFDS_DIFF_OUT_DCIEN: Differential Bidirectional Buffer with Complementary Outputs, Input Path Disable, and On-die Input Termination Disable
//
// UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

IOBUFDS_DIFF_OUT_DCIEN #(
    .DQS_BIAS("FALSE"),          // (FALSE, TRUE)
    .SIM_DEVICE("ULTRASCALE")   // Must be set to "ULTRASCALE"
)
IOBUFDS_DIFF_OUT_DCIEN_inst (
    .O(O),                       // 1-bit output: Buffer diff_p output
    .OB(OB),                     // 1-bit output: Buffer diff_n output
    .DCITERMDISABLE(DCITERMDISABLE), // 1-bit input: DCI Termination Disable
    .I(I),                       // 1-bit input: Buffer input
    .IBUFDISABLE(IBUFDISABLE),   // 1-bit input: Must be tied to a logic '0'
    .IO(IO),                    // 1-bit inout: Diff_p inout (connect directly to top-level port)
    .IOB(IOB),                  // 1-bit inout: Diff_n inout (connect directly to top-level port)
    .TM(TM),                    // 1-bit input: 3-state master enable input
    .TS(TS)                     // 1-bit input: 3-state slave enable input
);

// End of IOBUFDS_DIFF_OUT_DCIEN_inst instantiation
    
```

## For More Information

- See the *UltraScale Architecture SelectIO Resources User Guide* ([UG571](#)).
- See the [UltraScale Programmable Devices Documentation](#).

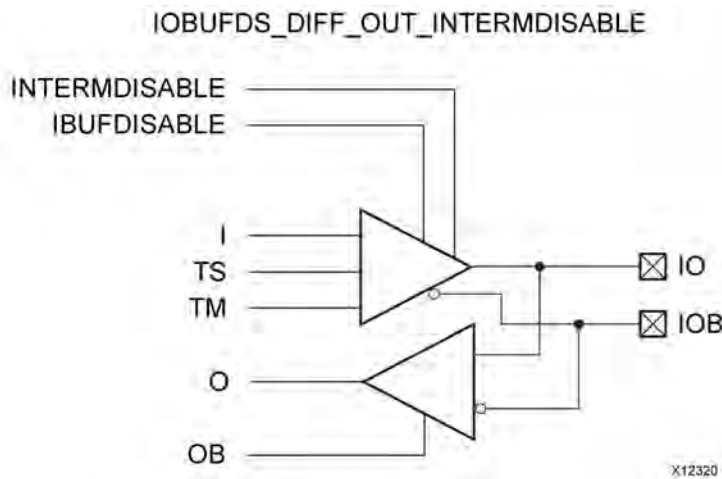
## IOBUFDS\_DIFF\_OUT\_INTERMDISABLE

Primitive: Differential Bidirectional Buffer with Complementary Outputs, Input Buffer Disable and On-die Input Termination Disable

PRIMITIVE\_GROUP: I/O

PRIMITIVE\_SUBGROUP: BIDIR\_BUFFER

Families: UltraScale, UltraScale+



### Introduction

The IOBUFDS\_DIFF\_OUT\_INTERMDISABLE primitive is available in the HR I/O banks. The IOBUFDS\_DIFF\_OUT\_INTERMDISABLE primitive has an INTERMDISABLE port that can be used to disable the optional on-die receiver termination feature. See UG571: UltraScale Architecture SelectIO Resources, "Uncalibrated Input Termination in I/O Banks" for more details on this feature. TM and TS must be connected to the same input (T) from the interconnect logic for this primitive to have the expected behavior that is specific to the UltraScale architecture.

The USE\_IBUFDISABLE attribute must be set to TRUE and SIM\_DEVICE to ULTRASCALE for the IOBUFDS\_DIFF\_OUT\_INTERMDISABLE primitive to have the expected behavior that is specific to the UltraScale architecture. If the I/O is using the on-die receiver termination features, this primitive disables the termination legs whenever the INTERMDISABLE signal is asserted High and the output buffer is 3-stated. When the output buffer is 3-stated (T = High), any on-die receiver termination is controlled by INTERMDISABLE. When the output buffer is not 3-stated (T = Low), the input buffer and on-die receiver termination are disabled and the O output (to the internal logic) is forced to a logic-Low.

I/O attributes that do not impact the logic function of the component, such as IOSTANDARD, DIFF\_TERM, and IBUF\_LOW\_PWR, should be supplied to the top-level port via an appropriate property. For details on applying such properties to the associated port, see UG912: Vivado Design Suite Properties Reference Guide. Attributes that impact the functionality, such as DQS\_BIAS, must be supplied to the component via a generic\_map (VHDL) or parameter (Verilog) in order to have the correct simulation behavior.

## Port Descriptions

Port	Direction	Width	Function
I	Input	1	Input of OBUF. Connect to the logic driving the output port.
IBUFDISABLE	Input	1	The IBUFDISABLE feature is not supported with this primitive in the UltraScale architecture. This port must be tied to logic '0'.
INTERMDISABLE	Input	1	Control to enable/disable on-chip input termination. This is generally used to reduce power in long periods of an idle state.
IO	Inout	1	Bidirectional diff_p port to be connected directly to top-level inout port.
IOB	Inout	1	Bidirectional diff_n port to be connected directly to top-level inout port.
O	Output	1	Output path of the buffer.
OB	Output	1	Output path of the buffer.
TM	Input	1	3-state enable input for the p-side or master side signifying whether the buffer acts as an input or output. This pin must be connected to the same signal as the TS input.
TS	Input	1	3-state enable input for the n-side or slave side signifying whether the buffer acts as an input or output. This pin must be connected to the same signal as the TM input.

## Design Entry Method

Instantiation	Recommended
Inference	No
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
DQS_BIAS	STRING	"FALSE", "TRUE"	"FALSE"	Provides pull-up/pull-down feature required for some DQS memory interface pins.
SIM_DEVICE	STRING	"ULTRASCALE", "7SERIES"	"7SERIES"	This attribute must be set to "ULTRASCALE" in order to exhibit the proper simulation behavior.
USE_IBUFDISABLE	STRING	"TRUE", "FALSE"	"TRUE"	This attribute must be unspecified or set to "TRUE" if specified.

## VHDL 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_DIFF_OUT_INTERMDISABLE: Differential Bidirectional Buffer with Complementary Outputs, Input Buffer Disable and On-die Input Termination Disable
--
-- UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

IOBUFDS_DIFF_OUT_INTERMDISABLE_inst : IOBUFDS_DIFF_OUT_INTERMDISABLE
generic map (
    DQS_BIAS => "FALSE",          -- (FALSE, TRUE)
    SIM_DEVICE => "ULTRASCALE"    -- Must be set to "ULTRASCALE"
)
port map (
    O => O,                        -- 1-bit output: Buffer diff_p output
    OB => OB,                       -- 1-bit output: Buffer diff_n output
    I => I,                          -- 1-bit input: Buffer input
    IBUFDISABLE => IBUFDISABLE,     -- 1-bit input: Must be tied to a logic '0'
    INTERMDISABLE => INTERMDISABLE, -- 1-bit input: Input Termination Disable
    IO => IO,                        -- 1-bit inout: Diff_p inout (connect directly to top-level port)
    IOB => IOB,                      -- 1-bit inout: Diff_n inout (connect directly to top-level port)
    TM => TM,                        -- 1-bit input: 3-state master enable input
    TS => TS                          -- 1-bit input: 3-state slave enable input
);

-- End of IOBUFDS_DIFF_OUT_INTERMDISABLE_inst instantiation
    
```

## Verilog Instantiation Template

```

// IOBUFDS_DIFF_OUT_INTERMDISABLE: Differential Bidirectional Buffer with Complementary Outputs, Input Buffer Disable and On-die Input Termination Disable
//
// UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

IOBUFDS_DIFF_OUT_INTERMDISABLE #(
    .DQS_BIAS("FALSE"),          // (FALSE, TRUE)
    .SIM_DEVICE("ULTRASCALE")    // Must be set to "ULTRASCALE"
)
IOBUFDS_DIFF_OUT_INTERMDISABLE_inst (
    .O(O),                        // 1-bit output: Buffer diff_p output
    .OB(OB),                      // 1-bit output: Buffer diff_n output
    .I(I),                          // 1-bit input: Buffer input
    .IBUFDISABLE(IBUFDISABLE),     // 1-bit input: Must be tied to a logic '0'
    .INTERMDISABLE(INTERMDISABLE), // 1-bit input: Input Termination Disable
    .IO(IO),                       // 1-bit inout: Diff_p inout (connect directly to top-level port)
    .IOB(IOB),                     // 1-bit inout: Diff_n inout (connect directly to top-level port)
    .TM(TM),                       // 1-bit input: 3-state master enable input
    .TS(TS)                        // 1-bit input: 3-state slave enable input
);

// End of IOBUFDS_DIFF_OUT_INTERMDISABLE_inst instantiation
    
```

## For More Information

- See the *UltraScale Architecture SelectIO Resources User Guide* ([UG571](#)).
- See the [UltraScale Programmable Devices Documentation](#).



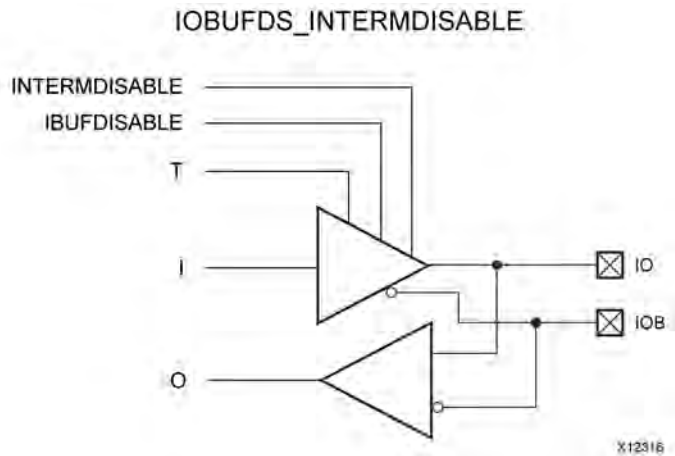
## IOBUFDS\_INTERMDISABLE

**Primitive: Differential Bidirectional Buffer With Input Buffer Disable and On-die Input**

PRIMITIVE\_GROUP: I/O

PRIMITIVE\_SUBGROUP: BIDIR\_BUFFER

Families: UltraScale, UltraScale+



### Introduction

The IOBUFDS\_INTERMDISABLE primitive is available in the HR I/O banks. It has an IBUFDISABLE port that can be used to disable the input buffer during periods when the buffer is not being used. The IOBUFDS\_INTERMDISABLE primitive also has an INTERMDISABLE port that can be used to disable the optional on-die receiver termination feature. See UG571: UltraScale Architecture SelectIO Resources, "Uncalibrated Input Termination in I/O Banks" for more details on this feature.

The IOBUFDS\_INTERMDISABLE primitive can disable the input buffer and force the O output to the internal logic to a logic-Low when the IBUFDISABLE signal is asserted High and the output buffer is 3-stated (T = High). The USE\_IBUFDISABLE attribute must be set to TRUE and SIM\_DEVICE to ULTRASCALE for this primitive to have the expected behavior that is specific to the UltraScale architecture. If the I/O is using the on-die receiver termination feature, this primitive disables the termination legs whenever the INTERMDISABLE signal is asserted High and the output buffer is 3-stated. When the output buffer is 3-stated (T = High), the input buffer and any on-die receiver termination are controlled by IBUFDISABLE and INTERMDISABLE, respectively. When the output buffer is not 3-stated (T = Low), the input buffer and on-die receiver termination are disabled and the O output (to the internal logic) is forced to a logic-Low. These features can be combined to reduce power whenever the input is idle for a period of time.

I/O attributes that do not impact the logic function of the component, such as IOSTANDARD, DIFF\_TERM, and IBUF\_LOW\_PWR, should be supplied to the top-level port via an appropriate property. For details on applying such properties to the associated port, see UG912: Vivado Design Suite Properties Reference Guide. Attributes that impact the functionality, such as DQS\_BIAS, must be supplied to the component via a generic\_map (VHDL) or parameter (Verilog) in order to have the correct simulation behavior.

## Port Descriptions

Port	Direction	Width	Function
I	Input	1	Input of OBUF. Connect to the logic driving the output port.
IBUFDISABLE	Input	1	Disables input path through the buffer and forces to a logic low. This feature is generally used to reduce power at times when the I/O is idle for a period of time.
INTERMDISABLE	Input	1	Control to enable/disable on-chip input termination. This is generally used to reduce power in long periods of an idle state.
IO	Inout	1	Bidirectional diff_p port to be connected directly to top-level inout port.
IOB	Inout	1	Bidirectional diff_n port to be connected directly to top-level inout port.
O	Output	1	Output path of the buffer.
T	Input	1	3-state enable input signifying whether the buffer acts as an input or output.

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
DQS_BIAS	STRING	"FALSE", "TRUE"	"FALSE"	Provides pull-up/pull-down feature required for some DQS memory interface pins.
SIM_DEVICE	STRING	"ULTRASCALE", "7SERIES"	"7SERIES"	This attribute must be set to "ULTRASCALE" in order to exhibit the proper simulation behavior.
USE_IBUFDISABLE	STRING	"TRUE", "FALSE"	"TRUE"	This attribute must be unspecified or set to "TRUE" if specified.

## VHDL 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_INTERMDISABLE: Differential Bidirectional Buffer With Input Buffer Disable and On-die Input
--                               UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

IOBUFDS_INTERMDISABLE_inst : IOBUFDS_INTERMDISABLE
generic map (
    DQS_BIAS => "FALSE",      -- (FALSE, TRUE)
    SIM_DEVICE => "ULTRASCALE" -- Must be set to "ULTRASCALE"
)
    
```

```

)
port map (
    O => O,           -- 1-bit output: Buffer output
    I => I,           -- 1-bit input: Buffer input
    IBUFDISABLE => IBUFDISABLE, -- 1-bit input: Buffer disable input, high=disable
    INTERMDISABLE => INTERMDISABLE, -- 1-bit input: Input Termination Disable
    IO => IO,         -- 1-bit inout: Diff_p inout (connect directly to top-level port)
    IOB => IOB,       -- 1-bit inout: Diff_n inout (connect directly to top-level port)
    T => T           -- 1-bit input: 3-state enable input
);

-- End of IOBUFDS_INTERMDISABLE_inst instantiation

```

## Verilog Instantiation Template

```

// IOBUFDS_INTERMDISABLE: Differential Bidirectional Buffer With Input Buffer Disable and On-die Input
//                               UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

IOBUFDS_INTERMDISABLE #(
    .DQS_BIAS("FALSE"), // (FALSE, TRUE)
    .SIM_DEVICE("ULTRASCALE") // Must be set to "ULTRASCALE"
)
IOBUFDS_INTERMDISABLE_inst (
    .O(O), // 1-bit output: Buffer output
    .I(I), // 1-bit input: Buffer input
    .IBUFDISABLE(IBUFDISABLE), // 1-bit input: Buffer disable input, high=disable
    .INTERMDISABLE(INTERMDISABLE), // 1-bit input: Input Termination Disable
    .IO(IO), // 1-bit inout: Diff_p inout (connect directly to top-level port)
    .IOB(IOB), // 1-bit inout: Diff_n inout (connect directly to top-level port)
    .T(T) // 1-bit input: 3-state enable input
);

// End of IOBUFDS_INTERMDISABLE_inst instantiation

```

## For More Information

- See the *UltraScale Architecture SelectIO Resources User Guide* ([UG571](#)).
- See the [UltraScale Programmable Devices Documentation](#).

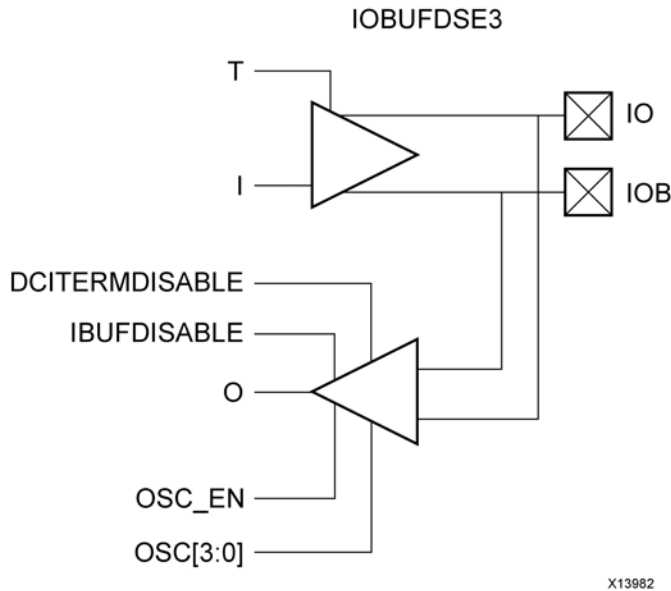
## IOBUFDSE3

Primitive: Differential Bidirectional I/O Buffer with Offset Calibration

PRIMITIVE\_GROUP: I/O

PRIMITIVE\_SUBGROUP: BIDIR\_BUFFER

Families: UltraScale, UltraScale+



### Introduction

The differential bidirectional input/output buffer primitive (IOBUFDSE3) is only supported in HP I/O banks. This UltraScale architecture specific primitive has functions similar to the IOBUFDSDCIEN along with controls for offset calibration with input buffer disable control (IBUFDISABLE) and on-die input termination disable control (DCITERMDISABLE) for the input buffer. The offset calibration feature is accessed using the OSC\_EN[1:0] and OSC[3:0] ports. The V<sub>REF</sub> scan feature is not supported with this primitive.

I/O attributes that do not impact the logic function of the component, such as IOSTANDARD, DIFF\_TERM, IBUF\_LOW\_PWR, and SLEW, should be supplied in XDC or to the top-level port via an appropriate property. For details on applying such properties to the associated port, see UG912: Vivado Design Suite Properties Reference Guide. Attributes that impact the functionality, such as DQS\_BIAS and SIM\_INPUT\_BUFFER\_OFFSET, must be supplied to the component via a generic\_map (VHDL) or parameter (Verilog) in order to have the correct simulation behavior.

### Logic Table

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

## Port Descriptions

Port	Direction	Width	Function
DCITERMDISABLE	Input	1	Control to enable/disable DCI termination. This is generally used to reduce power in long periods of an idle state.
I	Input	1	Input of OBUF. Connect to the logic driving the output port.
IBUFDISABLE	Input	1	Disables input path through the buffer and forces to a logic low. This feature is generally used to reduce power at times when the I/O is idle for a period of time.
IO	Inout	1	Bidirectional diff_p port to be connected directly to top-level inout port.
IOB	Inout	1	Bidirectional diff_n port to be connected directly to top-level inout port.
O	Output	1	Output path of the buffer.
OSC<3:0>	Input	4	Offset cancellation value
OSC_EN<1:0>	Input	2	Offset cancellation enable
T	Input	1	3-state enable input signifying whether the buffer acts as an input or output.

## Design Entry Method

Instantiation	Recommended
Inference	No
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
DQS_BIAS	STRING	"FALSE", "TRUE"	"FALSE"	Provides pull-up/pull-down feature required for some DQS memory interface pins.
SIM_INPUT_BUFFER_OFFSET	DECIMAL	-50 to 50	0	Offset value for simulation purposes.
USE_IBUFDISABLE	STRING	"FALSE", "TRUE"	"FALSE"	This attribute must be unspecified or set to "TRUE" if specified.

## VHDL Instantiation Template

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- IOBUFDSE3: Differential Bidirectional I/O Buffer with Offset Calibration
--           UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

IOBUFDSE3_inst : IOBUFDSE3
    
```

```

generic map (
    DQS_BIAS => "FALSE",          -- (FALSE, TRUE)
    SIM_INPUT_BUFFER_OFFSET => 0 -- Offset value for simulation (-50-50)
)
port map (
    O => O,                      -- 1-bit output: Buffer output
    DCITERMDISABLE => DCITERMDISABLE, -- 1-bit input: DCI Termination Disable
    I => I,                      -- 1-bit input: Buffer input
    IBUFDISABLE => IBUFDISABLE,  -- 1-bit input: Buffer disable input, high=disable
    IO => IO,                   -- 1-bit inout: Diff_p inout (connect directly to top-level port)
    IOB => IOB,                 -- 1-bit inout: Diff_n inout (connect directly to top-level port)
    OSC => OSC,                 -- 4-bit input: Offset cancellation value
    OSC_EN => OSC_EN,          -- 2-bit input: Offset cancellation enable
    T => T                      -- 1-bit input: 3-state enable input
);

-- End of IOBUFDSE3_inst instantiation
    
```

## Verilog Instantiation Template

```

// IOBUFDSE3: Differential Bidirectional I/O Buffer with Offset Calibration
//      UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

IOBUFDSE3 #(
    .DQS_BIAS("FALSE"),          // (FALSE, TRUE)
    .SIM_INPUT_BUFFER_OFFSET(0) // Offset value for simulation (-50-50)
)
IOBUFDSE3_inst (
    .O(O),                      // 1-bit output: Buffer output
    .DCITERMDISABLE(DCITERMDISABLE), // 1-bit input: DCI Termination Disable
    .I(I),                      // 1-bit input: Buffer input
    .IBUFDISABLE(IBUFDISABLE),  // 1-bit input: Buffer disable input, high=disable
    .IO(IO),                   // 1-bit inout: Diff_p inout (connect directly to top-level port)
    .IOB(IOB),                 // 1-bit inout: Diff_n inout (connect directly to top-level port)
    .OSC(OSC),                 // 4-bit input: Offset cancellation value
    .OSC_EN(OSC_EN),          // 2-bit input: Offset cancellation enable
    .T(T)                      // 1-bit input: 3-state enable input
);

// End of IOBUFDSE3_inst instantiation
    
```

## For More Information

- See the *UltraScale Architecture SelectIO Resources User Guide* ([UG571](#)).
- See the [UltraScale Programmable Devices Documentation](#).

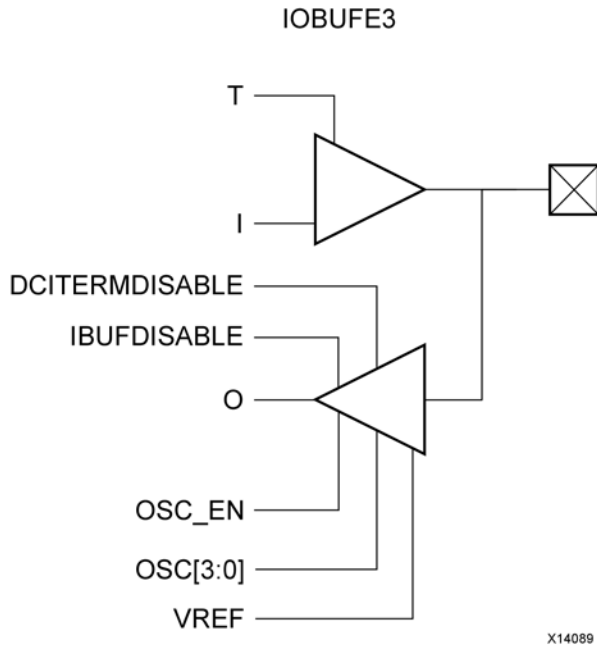
## IOBUFE3

Primitive: Bidirectional I/O Buffer with Offset Calibration and VREF Tuning

PRIMITIVE\_GROUP: I/O

PRIMITIVE\_SUBGROUP: BIDIR\_BUFFER

Families: UltraScale, UltraScale+



### Introduction

The bidirectional input/output buffer primitive (IOBUFE3) is only supported in HP I/O banks. This UltraScale architecture specific primitive has functions similar to the IOBUF\_DCIEN along with controls for offset calibration and  $V_{REF}$  tuning with input buffer disable (IBUFDISABLE) and on-die input termination control (DCITERMDISABLE) for the input buffer. The offset calibration feature is accessed using the OSC\_EN and OSC[3:0] ports. The  $V_{REF}$  scan feature is accessed using the HPIO\_VREF primitive in conjunction with IOBUFE3.

I/O attributes that do not impact the logic function of the component such as IOSTANDARD, DRIVE and IBUF\_LOW\_PWR should be supplied in XDC or to the top-level port via an appropriate property. For details on applying such properties to the associated port, see UG912: Vivado Design Suite Properties Reference Guide.

## Port Descriptions

Port	Direction	Width	Function
DCITERMDISABLE	Input	1	Control to enable/disable DCI termination. This is generally used to reduce power in long periods of an idle state.
I	Input	1	Input of OBUF. Connect to the logic driving the output port.
IBUFDISABLE	Input	1	Disables input path through the buffer and forces to a logic low. This feature is generally used to reduce power at times when the I/O is idle for a period of time.
IO	Inout	1	Bidirectional port to be connected directly to top-level inout port.
O	Output	1	Output path of the buffer.
OSC<3:0>	Input	4	Offset cancellation value
OSC_EN	Input	1	Offset cancellation enable
T	Input	1	3-state enable input signifying whether the buffer acts as an input or output.
VREF	Input	1	Vref input from HPIO_VREF

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
SIM_INPUT_BUFFER_OFFSET	DECIMAL	-50 to 50	0	Offset value for simulation purposes.
USE_IBUFDISABLE	STRING	"FALSE", "TRUE"	"FALSE"	This attribute must be unspecified or set to "TRUE" if specified.

## VHDL Instantiation Template

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- IOBUFE3: Bidirectional I/O Buffer with Offset Calibration and VREF Tuning
--           UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

IOBUFE3_inst : IOBUFE3
generic map (
    SIM_INPUT_BUFFER_OFFSET => 0 -- Offset value for simulation (-50-50)

```



```

)
port map (
    O => O,                -- 1-bit output: Buffer output
    DCITERMDISABLE => DCITERMDISABLE, -- 1-bit input: DCI Termination Disable
    I => I,                -- 1-bit input: Buffer input
    IBUFDISABLE => IBUFDISABLE, -- 1-bit input: Buffer disable input, high=disable
    IO => IO,              -- 1-bit inout: Buffer inout (connect directly to top-level port)
    OSC => OSC,            -- 4-bit input: Offset cancellation value
    OSC_EN => OSC_EN,     -- 1-bit input: Offset cancellation enable
    T => T,                -- 1-bit input: 3-state enable input
    VREF => VREF           -- 1-bit input: Vref input from HPIO_VREF
);

-- End of IOBUFE3_inst instantiation
    
```

## Verilog Instantiation Template

```

// IOBUFE3: Bidirectional I/O Buffer with Offset Calibration and VREF Tuning
// UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

IOBUFE3 #(
    .SIM_INPUT_BUFFER_OFFSET(0) // Offset value for simulation (-50-50)
)
IOBUFE3_inst (
    .O(O),                // 1-bit output: Buffer output
    .DCITERMDISABLE(DCITERMDISABLE), // 1-bit input: DCI Termination Disable
    .I(I),                // 1-bit input: Buffer input
    .IBUFDISABLE(IBUFDISABLE), // 1-bit input: Buffer disable input, high=disable
    .IO(IO),              // 1-bit inout: Buffer inout (connect directly to top-level port)
    .OSC(OSC),            // 4-bit input: Offset cancellation value
    .OSC_EN(OSC_EN),     // 1-bit input: Offset cancellation enable
    .T(T),                // 1-bit input: 3-state enable input
    .VREF(VREF)           // 1-bit input: Vref input from HPIO_VREF
);

// End of IOBUFE3_inst instantiation
    
```

## For More Information

- See the *UltraScale Architecture SelectIO Resources User Guide* ([UG571](#)).
- See the [UltraScale Programmable Devices Documentation](#).

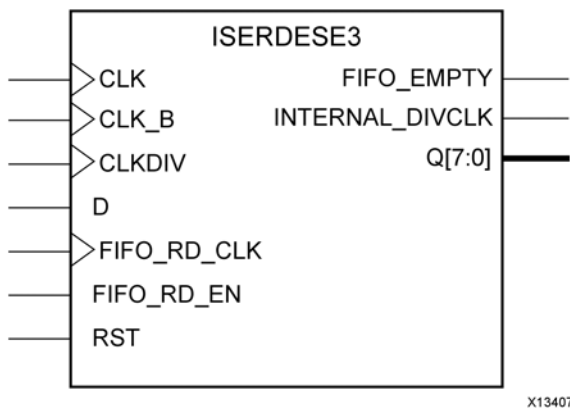
# ISERDESE3

## Primitive: Input SERIAL/DESerializer

PRIMITIVE\_GROUP: I/O

PRIMITIVE\_SUBGROUP: SERDES

Families: UltraScale, UltraScale+



## Introduction

In component mode, the ISERDESE3 in UltraScale devices is a dedicated serial-to-parallel converter with specific clocking and logic features designed to facilitate the implementation of high-speed source-synchronous applications. The ISERDESE3 avoids the additional timing complexities encountered when designing deserializers in the device fabric. ISERDESE3 features include: Dedicated Deserializer/Serial-to-Parallel Converter, which enables high-speed data transfer without requiring the device fabric to match the input data frequency. This converter supports both single data rate (SDR) and double data rate (DDR) modes. In SDR mode, the serial-to-parallel converter creates a 4-bit wide parallel word by retrieving data from every other Q pin. In DDR mode, the serial-to-parallel converter creates an 8-bit-wide parallel word.

## Port Descriptions

Port	Direction	Width	Function
CLK	Input	1	The high-speed clock input (CLK) is used to clock in the input serial data stream.
CLK_B	Input	1	The inverted high-speed clock input
CLKDIV	Input	1	The divided clock input (CLKDIV) is typically a divided version of CLK (depending on the width of the implemented deserialization). It drives the output of the serial-to-parallel converter and the CE module.
D	Input	1	The serial input data port (D) is the serial (high-speed) data input port of the ISERDESE3. This port accepts data from the IOB or device Fabric.
FIFO_EMPTY	Output	1	FIFO empty flag
FIFO_RD_CLK	Input	1	FIFO read clock

Port	Direction	Width	Function
FIFO_RD_EN	Input	1	Enables reading the FIFO when asserted
INTERNAL_DIVCLK	Output	1	Internally divided down clock used to launch data from ISERDES to fabric when FIFO is disabled (do not connect)
Q<7:0>	Output	8	8-bit registered output
RST	Input	1	Asynchronous Reset, active level based on IS_RST_INVERTED

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	Yes

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
DATA_WIDTH	DECIMAL	8, 4	8	Defines the width of the serial-to-parallel converter. Running in SDR mode is done by retrieving data from every other Q pin.
FIFO_ENABLE	STRING	"FALSE", "TRUE"	"FALSE"	Setting FIFO_ENABLE to TRUE uses the FIFO and setting FIFO_ENABLE to FALSE bypasses the FIFO
FIFO_SYNC_MODE	STRING	"FALSE", "TRUE"	"FALSE"	Set FIFO_SYNC_MODE to TRUE when the internal write clock and the FIFO_RD_CLK are coming from a common source and all clock timing from write to read must be met. Set FIFO_SYNC_MODE to FALSE will cause the internal 2-stage flop synchronizers to be used and will add latency to when FIFO_EMPTY will deassert after data is first written to the FIFO.
IDDR_MODE	STRING	"FALSE", "TRUE"	"FALSE"	Internal property for Vivado primitive mapping. Do not modify.
IS_CLK_B_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether the CLK_B pin is active-low or active-high.
IS_CLK_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether the CLK pin is active-high or active-low.
IS_RST_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether the RST pin is active-high or active-low.
SIM_DEVICE	STRING	"ULTRASCALE", "ULTRASCALE_PLUS_ES1"	"ULTRASCALE"	Set the device version

## VHDL Instantiation Template

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- ISERDESE3: Input SERIAL/DESerializer
-- UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

ISERDESE3_inst : ISERDESE3
generic map (
    DATA_WIDTH => 8,                -- Parallel data width (4,8)
    FIFO_ENABLE => "FALSE",         -- Enables the use of the FIFO
    FIFO_SYNC_MODE => "FALSE",     -- Enables the use of internal 2-stage synchronizers on the FIFO
    IS_CLK_B_INVERTED => '0',       -- Optional inversion for CLK_B
    IS_CLK_INVERTED => '0',        -- Optional inversion for CLK
    IS_RST_INVERTED => '0',        -- Optional inversion for RST
    SIM_DEVICE => "ULTRASCALE"     -- Set the device version (ULTRASCALE, ULTRASCALE_PLUS_ES1)
)
port map (
    FIFO_EMPTY => FIFO_EMPTY,      -- 1-bit output: FIFO empty flag
    INTERNAL_DIVCLK => INTERNAL_DIVCLK, -- 1-bit output: Internally divided down clock used when FIFO is
                                        -- disabled (do not connect)

    Q => Q,                          -- 8-bit registered output
    CLK => CLK,                        -- 1-bit input: High-speed clock
    CLKDIV => CLKDIV,                 -- 1-bit input: Divided Clock
    CLK_B => CLK_B,                  -- 1-bit input: Inversion of High-speed clock CLK
    D => D,                          -- 1-bit input: Serial Data Input
    FIFO_RD_CLK => FIFO_RD_CLK,      -- 1-bit input: FIFO read clock
    FIFO_RD_EN => FIFO_RD_EN,       -- 1-bit input: Enables reading the FIFO when asserted
    RST => RST                        -- 1-bit input: Asynchronous Reset
);

-- End of ISERDESE3_inst instantiation
    
```

## Verilog Instantiation Template

```

// ISERDESE3: Input SERIAL/DESerializer
//      UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

ISERDESE3 #(
    .DATA_WIDTH(8),           // Parallel data width (4,8)
    .FIFO_ENABLE("FALSE"),   // Enables the use of the FIFO
    .FIFO_SYNC_MODE("FALSE"), // Enables the use of internal 2-stage synchronizers on the FIFO
    .IS_CLK_B_INVERTED(1'b0), // Optional inversion for CLK_B
    .IS_CLK_INVERTED(1'b0),  // Optional inversion for CLK
    .IS_RST_INVERTED(1'b0),  // Optional inversion for RST
    .SIM_DEVICE("ULTRASCALE") // Set the device version (ULTRASCALE, ULTRASCALE_PLUS_ES1)
)
ISERDESE3_inst (
    .FIFO_EMPTY(FIFO_EMPTY), // 1-bit output: FIFO empty flag
    .INTERNAL_DIVCLK(INTERNAL_DIVCLK), // 1-bit output: Internally divided down clock used when FIFO is
                                        // disabled (do not connect)

    .Q(Q), // 8-bit registered output
    .CLK(CLK), // 1-bit input: High-speed clock
    .CLKDIV(CLKDIV), // 1-bit input: Divided Clock
    .CLK_B(CLK_B), // 1-bit input: Inversion of High-speed clock CLK
    .D(D), // 1-bit input: Serial Data Input
    .FIFO_RD_CLK(FIFO_RD_CLK), // 1-bit input: FIFO read clock
    .FIFO_RD_EN(FIFO_RD_EN), // 1-bit input: Enables reading the FIFO when asserted
    .RST(RST) // 1-bit input: Asynchronous Reset
);

// End of ISERDESE3_inst instantiation

```

## For More Information

- See the *UltraScale Architecture SelectIO Resources User Guide* ([UG571](#)).
- See the [UltraScale Programmable Devices Documentation](#).

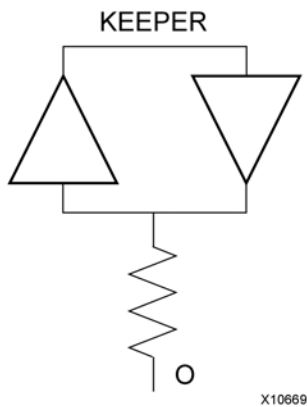
# KEEPER

Primitive: I/O Weak Keeper

PRIMITIVE\_GROUP: I/O

PRIMITIVE\_SUBGROUP: WEAK\_DRIVER

Families: UltraScale, UltraScale+



## Introduction

The design element is a weak keeper element that retains the value of the I/O when not being driven. For example, if a logic 1 is being driven onto the I/O, KEEPER drives a weak/resistive 1 onto the pin/port. If the net driver is then 3-stated, KEEPER continues to drive a weak/resistive 1 onto the pin/port.

## Port Descriptions

Port	Direction	Width	Function
O	Inout	1	Keeper output. Connect directly to a top_level port.

## Design Entry Method

Instantiation	Yes
Inference	Yes, via property
IP and IP Integrator Catalog	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 Weak Keeper
--      UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

KEEPER_inst : KEEPER
port map (
    O => O -- 1-bit inout: Keeper output (connect directly to top-level port)
);

-- End of KEEPER_inst instantiation
```

## Verilog Instantiation Template

```
// KEEPER: I/O Weak Keeper
//      UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

KEEPER KEEPER_inst (
    .O(O) // 1-bit inout: Keeper output (connect directly to top-level port)
);

// End of KEEPER_inst instantiation
```

## For More Information

- See the *UltraScale Architecture SelectIO Resources User Guide* ([UG571](#)).
- See the [UltraScale Programmable Devices Documentation](#).

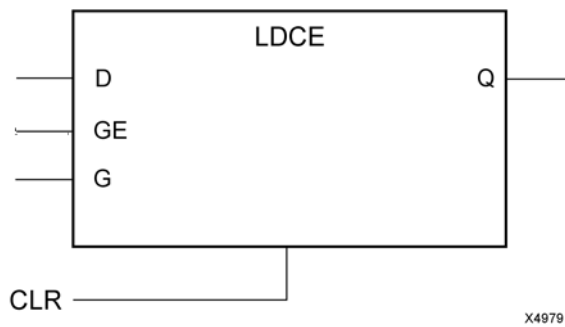
## LDCE

Primitive: Transparent Latch with Clock Enable and Asynchronous Clear

PRIMITIVE\_GROUP: REGISTER

PRIMITIVE\_SUBGROUP: LATCH

Families: UltraScale, UltraScale+



### Introduction

This design element is a transparent data latch with asynchronous clear and gate enable. When the asynchronous clear input (CLR) is active, 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 active and CLR is not active. If (GE) is Low, data on (D) cannot be latched. The data on the (D) input during the 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 initialized when power is applied. When global set/reset (GSR) is active upon power-up or when asserted via the STARTUP block, the value of the INIT attribute is placed on the latch's output.

### 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



## Port Descriptions

Port	Direction	Width	Function
CLR	Input	1	Asynchronous clear. Polarity is determined by the IS_CLR_INVERTED attribute.
D	Input	1	Data input
G	Input	1	Gate input. Polarity is determined by the IS_G_INVERTED attribute.
GE	Input	1	Active-High latch gate enable
Q	Output	1	Data output

## Design Entry Method

Instantiation	Yes
Inference	Yes
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	BINARY	1'b0, 1'b1	1'b0	Sets the initial value of Q output after configuration or when GSR is asserted via the STARTUP block.
<b>Programmable Inversion Attributes:</b> Specifies whether or not to use the optional inversion on specific pins of this component to change the active polarity of the pin function. When set to 1 on a gate pin (G), it creates an active-Low latch. When set to 1 on other pins, it changes the function to behave active-Low rather than active-High. If an external inverter is specified on one of these associated pins, the Vivado Design Suite will automatically set this attribute during the opt_design stage so that additional logic is not necessary for changing the input polarity.				
IS_CLR_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion for the CLR pin of this component.
IS_G_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion for the G pin of this component.

## VHDL Instantiation Template

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- LDCE: Transparent Latch with Clock Enable and Asynchronous Clear
--      UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

LDCE_inst : LDCE
generic map (
    INIT => '0',           -- Initial value of latch, '0', '1'
    -- Programmable Inversion Attributes: Specifies the use of the built-in programmable inversion
    IS_CLR_INVERTED => '0', -- Optional inversion for CLR

```

```

    IS_G_INVERTED => '0'    -- Optional inversion for G
  )
  port map (
    Q => Q,      -- 1-bit output: Data
    CLR => CLR,  -- 1-bit input: Asynchronous clear
    D => D,      -- 1-bit input: Data
    G => G,      -- 1-bit input: Gate
    GE => GE     -- 1-bit input: Gate enable
  );

-- End of LDCE_inst instantiation

```

## Verilog Instantiation Template

```

// LDCE: Transparent Latch with Clock Enable and Asynchronous Clear
//      UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

LDCE #(
  .INIT(1'b0),           // Initial value of latch, 1'b0, 1'b1
  // Programmable Inversion Attributes: Specifies the use of the built-in programmable inversion
  .IS_CLR_INVERTED(1'b0), // Optional inversion for CLR
  .IS_G_INVERTED(1'b0)  // Optional inversion for G
)
LDCE_inst (
  .Q(Q),      // 1-bit output: Data
  .CLR(CLR), // 1-bit input: Asynchronous clear
  .D(D),      // 1-bit input: Data
  .G(G),      // 1-bit input: Gate
  .GE(GE)    // 1-bit input: Gate enable
);

// End of LDCE_inst instantiation

```

## For More Information

- See the *UltraScale Architecture Migration: Methodology Guide* ([UG1026](#)).
- See the [UltraScale Programmable Devices Documentation](#).

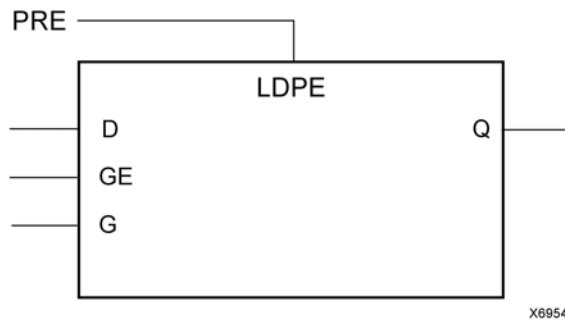
## LDPE

### Primitive: Transparent Latch with Clock Enable and Asynchronous Preset

PRIMITIVE\_GROUP: REGISTER

PRIMITIVE\_SUBGROUP: LATCH

Families: UltraScale, UltraScale+



## Introduction

This design element is a transparent data latch with asynchronous preset and gate enable. When the asynchronous preset input (PRE) is active, it overrides the other inputs and presets the data (Q) output High. Q reflects the data (D) input while the gate (G) input and gate enable (GE) are active and PRE is not active. If (GE) is Low, data on (D) cannot be latched. The data on the (D) input during the 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 initialized when power is applied. When global set/reset (GSR) is active upon power-up or when asserted via the STARTUP block, the value of the INIT attribute is placed on the latch's output.

## 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

## Port Descriptions

Port	Direction	Width	Function
D	Input	1	Data input
G	Input	1	Gate input. Polarity is determined by the IS_G_INVERTED attribute.
GE	Input	1	Active-High latch gate enable
PRE	Input	1	Asynchronous preset. Polarity is determined by the IS_PRE_INVERTED attribute.
Q	Output	1	Data output

## Design Entry Method

Instantiation	Yes
Inference	Yes
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	BINARY	1'b1, 1'b0	1'b1	Sets the initial value of Q output after configuration or when GSR is asserted via the STARTUP block.
<b>Programmable Inversion Attributes:</b> Specifies whether or not to use the optional inversion on specific pins of this component to change the active polarity of the pin function. When set to 1 on a gate pin (G), it creates an active-Low latch. When set to 1 on other pins, it changes the function to behave active-Low rather than active-High. If an external inverter is specified on one of these associated pins, the Vivado Design Suite will automatically set this attribute during the opt_design stage so that additional logic is not necessary for changing the input polarity.				
IS_G_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion for the G pin of this component.
IS_PRE_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion for the PRE pin of this component

## VHDL Instantiation Template

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- LDPE: Transparent Latch with Clock Enable and Asynchronous Preset
--      UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

LDPE_inst : LDPE
generic map (
    INIT => '1',           -- Initial value of latch, '0', '1'
    -- Programmable Inversion Attributes: Specifies the use of the built-in programmable inversion
    IS_G_INVERTED => '0',  -- Optional inversion for G

```

```

        IS_PRE_INVERTED => '0' -- Optional inversion for PRE
    )
    port map (
        Q => Q,      -- 1-bit output: Data
        D => D,      -- 1-bit input: Data
        G => G,      -- 1-bit input: Gate
        GE => GE,    -- 1-bit input: Gate enable
        PRE => PRE   -- 1-bit input: Asynchronous preset
    );

-- End of LDPE_inst instantiation

```

## Verilog Instantiation Template

```

// LDPE: Transparent Latch with Clock Enable and Asynchronous Preset
//      UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

LDPE #(
    .INIT(1'b1),           // Initial value of latch, 1'b0, 1'b1
    // Programmable Inversion Attributes: Specifies the use of the built-in programmable inversion
    .IS_G_INVERTED(1'b0), // Optional inversion for G
    .IS_PRE_INVERTED(1'b0) // Optional inversion for PRE
)
LDPE_inst (
    .Q(Q),      // 1-bit output: Data
    .D(D),      // 1-bit input: Data
    .G(G),      // 1-bit input: Gate
    .GE(GE),    // 1-bit input: Gate enable
    .PRE(PRE)   // 1-bit input: Asynchronous preset
);

// End of LDPE_inst instantiation

```

## For More Information

- See the *UltraScale Architecture Migration: Methodology Guide* ([UG1026](#)).
- See the [UltraScale Programmable Devices Documentation](#).

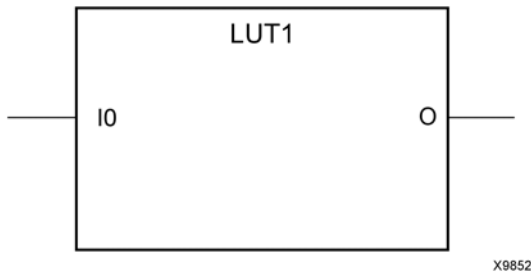
# LUT1

## Primitive: 1-Bit Look-Up Table

PRIMITIVE\_GROUP: CLB

PRIMITIVE\_SUBGROUP: LUT

Families: UltraScale, UltraScale+



## Introduction

This design element is a 1-bit look-up table (LUT). This element provides a look-up table version of a buffer or inverter.

The INIT parameter for the 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 or generics 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 or generics.

A LUT1 may be grouped with another LUT1, LUT2, LUT3 or LUT4 and placed into a single LUT6 resource. It may also be placed with a LUT5 however it must share a common input signal. The Vivado Design Suite will automatically combine LUTs when necessary or advantageous. However, this can be manually controlled by specifying a LUTNM or HLUTNM on the associated LUT components to specify specific grouping within a single LUT resource.

## 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
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	HEX	2'h0 to 2'h3	2'h0	Specifies the 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;

-- LUT1: 1-Bit Look-Up Table
-- UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

LUT1_inst : LUT1
generic map (
    INIT => X"0" -- Logic function
)
port map (
    O => O, -- 1-bit output: LUT
    I0 => I0 -- 1-bit input: LUT
);

-- End of LUT1_inst instantiation
    
```

## Verilog Instantiation Template

```

// LUT1: 1-Bit Look-Up Table
// UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

LUT1 #(
    .INIT(2'h0) // Logic function
)
LUT1_inst (
    .O(O), // 1-bit output: LUT
    .I0(I0) // 1-bit input: LUT
);

// End of LUT1_inst instantiation
    
```

## For More Information

- See the *UltraScale Architecture Configurable Logic Block User Guide* ([UG574](#)).
- See the [UltraScale Programmable Devices Documentation](#).

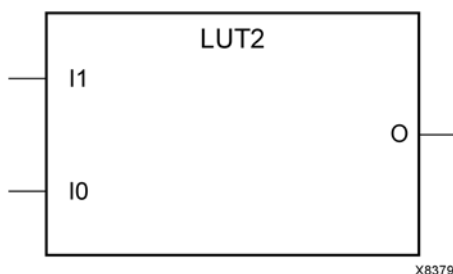
## LUT2

### Primitive: 2-Bit Look-Up Table

PRIMITIVE\_GROUP: CLB

PRIMITIVE\_SUBGROUP: LUT

Families: UltraScale, UltraScale+



### Introduction

This design element is a 2-bit look-up table (LUT). This element allows the creation of any logical function with two inputs.

The INIT parameter for the 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 or generics 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 or generics.

A LUT2 may be grouped with a LUT1, LUT2, LUT3, LUT4 or LUT5 and placed into a single LUT6 resource as long as the combined input signals do not exceed 5 unique inputs. The Vivado Design Suite will automatically combine LUTs when necessary or advantageous. However this can be manually controlled by specifying a LUTNM or HLUTNM on the associated LUT components to specify specific grouping within a single LUT resource.



## 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
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	HEX	4'h0 to 4'hf	4'h0	Specifies the 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;

-- LUT2: 2-Bit Look-Up Table
--      UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

LUT2_inst : LUT2
generic map (
    INIT => X"0" -- Logic function
)
port map (
    O => O, -- 1-bit output: LUT
    I0 => I0, -- 1-bit input: LUT
    I1 => I1 -- 1-bit input: LUT
);

-- End of LUT2_inst instantiation
    
```

## Verilog Instantiation Template

```
// LUT2: 2-Bit Look-Up Table
//      UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

LUT2 #(
    .INIT(4'h0) // Logic function
)
LUT2_inst (
    .O(O), // 1-bit output: LUT
    .I0(I0), // 1-bit input: LUT
    .I1(I1) // 1-bit input: LUT
);

// End of LUT2_inst instantiation
```

## For More Information

- See the *UltraScale Architecture Configurable Logic Block User Guide* ([UG574](#)).
- See the [UltraScale Programmable Devices Documentation](#).

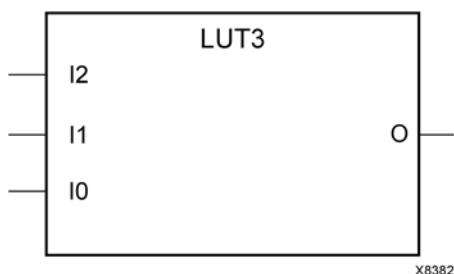
## LUT3

### Primitive: 3-Bit Look-Up Table

PRIMITIVE\_GROUP: CLB

PRIMITIVE\_SUBGROUP: LUT

Families: UltraScale, UltraScale+



## Introduction

This design element is a 3-bit look-up table (LUT). This element allows the creation of any logical function with three inputs.

The INIT parameter for the 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 or generics 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 or generics.

A LUT3 may be grouped with a LUT1, LUT2, LUT3, LUT4 or LUT5 and placed into a single LUT6 resource as long as the combined input signals do not exceed 5 unique inputs. The Vivado Design Suite will automatically combine LUTs when necessary or advantageous. However this can be manually controlled by specifying a LUTNM or HLUTNM on the associated LUT components to specify specific grouping within a single LUT resource.

## 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
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	HEX	8'h00 to 8'hff	8'h00	Specifies the 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;

-- LUT3: 3-Bit Look-Up Table
-- UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

LUT3_inst : LUT3
generic map (
    INIT => X"00" -- Logic function
)
port map (
    O => O, -- 1-bit output: LUT
    I0 => I0, -- 1-bit input: LUT
    I1 => I1, -- 1-bit input: LUT
    I2 => I2 -- 1-bit input: LUT
);

-- End of LUT3_inst instantiation
    
```

## Verilog Instantiation Template

```
// LUT3: 3-Bit Look-Up Table
//      UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

LUT3 #(
    .INIT(8'h00) // Logic function
)
LUT3_inst (
    .O(O), // 1-bit output: LUT
    .I0(I0), // 1-bit input: LUT
    .I1(I1), // 1-bit input: LUT
    .I2(I2) // 1-bit input: LUT
);

// End of LUT3_inst instantiation
```

## For More Information

- See the *UltraScale Architecture Configurable Logic Block User Guide* ([UG574](#)).
- See the [UltraScale Programmable Devices Documentation](#).

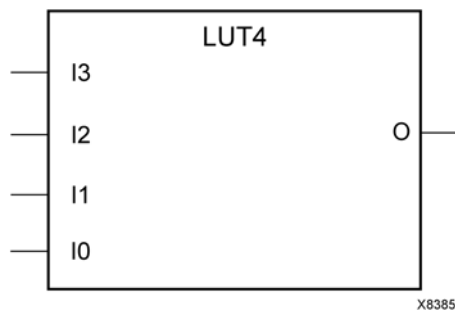
## LUT4

### Primitive: 4-Bit Look-Up Table

PRIMITIVE\_GROUP: CLB

PRIMITIVE\_SUBGROUP: LUT

Families: UltraScale, UltraScale+



## Introduction

This design element is a 4-bit look-up table (LUT). This element allows the creation of any logical function with four inputs.

The INIT parameter for the 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 or generics 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 or generics.

A LUT4 may be grouped with a LUT1, LUT2, LUT3, LUT4 or LUT5 and placed into a single LUT6 resource as long as the combined input signals do not exceed 5 unique inputs. The Vivado Design Suite will automatically combine LUTs when necessary or advantageous. However this can be manually controlled by specifying a LUTNM or HLUTNM on the associated LUT components to specify specific grouping within a single LUT resource.

## Logic Table

Inputs				Outputs
I3	I2	I1	I0	O
0	0	0	0	INIT[0]
0	0	0	1	INIT[1]
0	0	1	0	INIT[2]
0	0	1	1	INIT[3]
0	1	0	0	INIT[4]
0	1	0	1	INIT[5]
0	1	1	0	INIT[6]
0	1	1	1	INIT[7]
1	0	0	0	INIT[8]
1	0	0	1	INIT[9]
1	0	1	0	INIT[10]
1	0	1	1	INIT[11]
1	1	0	0	INIT[12]
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
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	HEX	16'h0000 to 16'hffff	16'h0000	Specifies the 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;

-- LUT4: 4-Bit Look-Up Table
--      UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

LUT4_inst : LUT4
generic map (
  INIT => X"0000" -- Logic function
)
port map (
  O => O, -- 1-bit output: LUT
  I0 => I0, -- 1-bit input: LUT
  I1 => I1, -- 1-bit input: LUT
  I2 => I2, -- 1-bit input: LUT
  I3 => I3 -- 1-bit input: LUT
);

-- End of LUT4_inst instantiation
  
```

## Verilog Instantiation Template

```

// LUT4: 4-Bit Look-Up Table
//      UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

LUT4 #(
  .INIT(16'h0000) // Logic function
)
LUT4_inst (
  .O(O), // 1-bit output: LUT
  .I0(I0), // 1-bit input: LUT
  .I1(I1), // 1-bit input: LUT
  .I2(I2), // 1-bit input: LUT
  .I3(I3) // 1-bit input: LUT
);

// End of LUT4_inst instantiation
  
```

## For More Information

- See the *UltraScale Architecture Configurable Logic Block User Guide* ([UG574](#)).
- See the [UltraScale Programmable Devices Documentation](#).



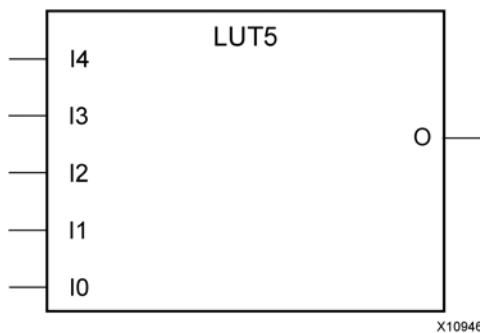
## LUT5

### Primitive: 5-Bit Look-Up Table

PRIMITIVE\_GROUP: CLB

PRIMITIVE\_SUBGROUP: LUT

Families: UltraScale, UltraScale+



## Introduction

This design element is a 5-bit look-up table (LUT). This element allows the creation of any logical function with five inputs.

The INIT parameter for the 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 or generics 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 or generics.

A LUT5 may be grouped with a LUT1, LUT2, LUT3, LUT4 or LUT5 and placed into a single LUT6 resource as long as the combined input signals do not exceed 5 unique inputs. The Vivado Design Suite will automatically combine LUTs when necessary or advantageous. However this can be manually controlled by specifying a LUTNM or HLUTNM on the associated LUT components to specify specific grouping within a single LUT resource.

## 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]
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

Port	Direction	Width	Function
I0	Input	1	LUT input
I1	Input	1	LUT input
I2	Input	1	LUT input
I3	Input	1	LUT input
I4	Input	1	LUT input
O	Output	1	LUT output

## Design Entry Method

Instantiation	Yes
Inference	Recommended
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	HEX	Any 32-bit HEX value	All zeroes	Specifies the 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;

-- LUT5: 5-Bit Look-Up Table
-- UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

LUT5_inst : LUT5
generic map (
    INIT => X"00000000" -- Logic function
)
port map (
    O => O, -- 1-bit output: LUT
    I0 => I0, -- 1-bit input: LUT
    I1 => I1, -- 1-bit input: LUT
    I2 => I2, -- 1-bit input: LUT
    I3 => I3, -- 1-bit input: LUT
    I4 => I4 -- 1-bit input: LUT
);

-- End of LUT5_inst instantiation
    
```

## Verilog Instantiation Template

```
// LUT5: 5-Bit Look-Up Table
//      UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

LUT5 #(
    .INIT(32'h00000000) // Logic function
)
LUT5_inst (
    .O(O), // 1-bit output: LUT
    .I0(I0), // 1-bit input: LUT
    .I1(I1), // 1-bit input: LUT
    .I2(I2), // 1-bit input: LUT
    .I3(I3), // 1-bit input: LUT
    .I4(I4) // 1-bit input: LUT
);

// End of LUT5_inst instantiation
```

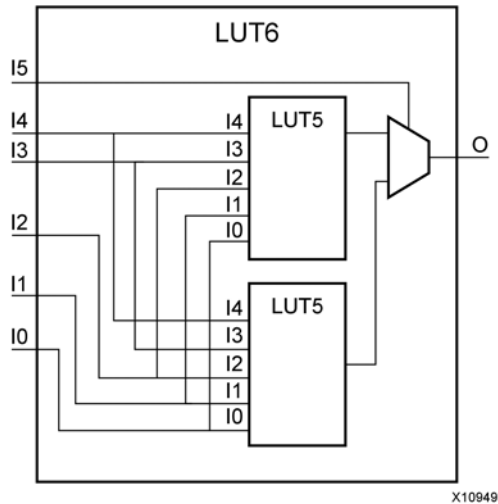
## For More Information

- See the *UltraScale Architecture Configurable Logic Block User Guide* ([UG574](#)).
- See the [UltraScale Programmable Devices Documentation](#).

# LUT6

## Primitive: 6-Bit Look-Up Table

PRIMITIVE\_GROUP: CLB  
 PRIMITIVE\_SUBGROUP: LUT  
 Families: UltraScale, UltraScale+



## Introduction

This design element is a 6-bit look-up table (LUT). This element allows the creation of any logical function with six inputs.

The INIT parameter for the 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 or generics 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 or generics.

## Logic Table

Inputs						Outputs
I15	I14	I13	I12	I11	I10	O
0	0	0	0	0	0	INIT[0]
0	0	0	0	0	1	INIT[1]

Inputs						Outputs
I5	I4	I3	I2	I1	I0	O
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]

Inputs						Outputs
I5	I4	I3	I2	I1	I0	O
1	0	0	1	1	0	INIT[38]
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

Port	Direction	Width	Function
I0	Input	1	LUT input
I1	Input	1	LUT input
I2	Input	1	LUT input
I3	Input	1	LUT input
I4	Input	1	LUT input

Port	Direction	Width	Function
I5	Input	1	LUT input
O	Output	1	LUT output

## Design Entry Method

Instantiation	Yes
Inference	Recommended
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	HEX	Any 64-bit HEX value	All zeroes	Specifies the 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;

-- LUT6: 6-Bit Look-Up Table
--      UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

LUT6_inst : LUT6
generic map (
    INIT => X"0000000000000000" -- Logic function
)
port map (
    O => O, -- 1-bit output: LUT
    I0 => I0, -- 1-bit input: LUT
    I1 => I1, -- 1-bit input: LUT
    I2 => I2, -- 1-bit input: LUT
    I3 => I3, -- 1-bit input: LUT
    I4 => I4, -- 1-bit input: LUT
    I5 => I5 -- 1-bit input: LUT
);

-- End of LUT6_inst instantiation
    
```



## Verilog Instantiation Template

```
// LUT6: 6-Bit Look-Up Table
//      UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

LUT6 #(
    .INIT(64'h0000000000000000) // Logic function
)
LUT6_inst (
    .O(O), // 1-bit output: LUT
    .I0(I0), // 1-bit input: LUT
    .I1(I1), // 1-bit input: LUT
    .I2(I2), // 1-bit input: LUT
    .I3(I3), // 1-bit input: LUT
    .I4(I4), // 1-bit input: LUT
    .I5(I5) // 1-bit input: LUT
);

// End of LUT6_inst instantiation
```

## For More Information

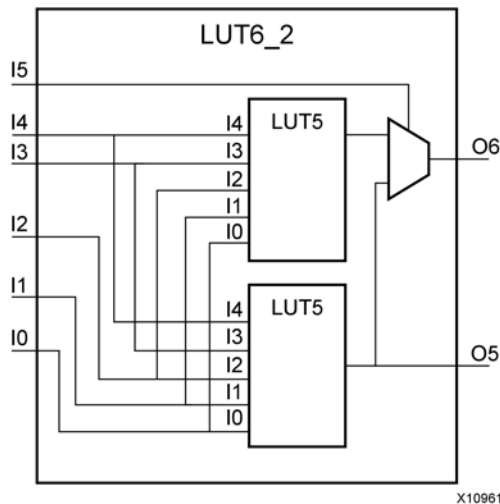
- See the *UltraScale Architecture Configurable Logic Block User Guide* ([UG574](#)).
- See the [UltraScale Programmable Devices Documentation](#).

# LUT6\_2

Primitive: Six-input, 2-output, Look-Up Table

PRIMITIVE\_GROUP: CLB

PRIMITIVE\_SUBGROUP: LUT



## 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 eight look-up tables in the CLB.

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'hffffffffffff (X"FFFFFFFFFFFFFFFF" 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]
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]

Inputs						Outputs	
I5	I4	I3	I2	I1	I0	O5	O6
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]
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 Description

Port	Direction	Width	Function
I0	Input	1	LUT input
I1	Input	1	LUT input
I2	Input	1	LUT input
I3	Input	1	LUT input
I4	Input	1	LUT input
I5	Input	1	LUT input
O5	Output	1	6/5-LUT output
O6	Output	1	5-LUT output

## Design Entry Method

Instantiation	Yes
Inference	Recommended
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	HEX	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
-- UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

LUT6_2_inst : LUT6_2
generic map (
    INIT => X"0000000000000000") -- Specify LUT Contents
port map (
    O6 => O6, -- 6/5-LUT output (1-bit)
    O5 => O5, -- 5-LUT output (1-bit)
    I0 => I0, -- LUT input (1-bit)
    I1 => I1, -- LUT input (1-bit)
    I2 => I2, -- LUT input (1-bit)
    I3 => I3, -- LUT input (1-bit)
    I4 => I4, -- LUT input (1-bit)
    I5 => I5 -- LUT input (1-bit)
);

-- End of LUT6_2_inst instantiation
    
```

## Verilog Instantiation Template

```
// LUT6_2: 6-input, 2 output Look-Up Table
//      UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

LUT6_2 #(
    .INIT(64'h0000000000000000) // Specify LUT Contents
) LUT6_2_inst (
    .O6(O6), // 1-bit LUT6 output
    .O5(O5), // 1-bit lower LUT5 output
    .I0(I0), // 1-bit LUT input
    .I1(I1), // 1-bit LUT input
    .I2(I2), // 1-bit LUT input
    .I3(I3), // 1-bit LUT input
    .I4(I4), // 1-bit LUT input
    .I5(I5) // 1-bit LUT input (fast MUX select only available to O6 output)
);

// End of LUT6_2_inst instantiation
```

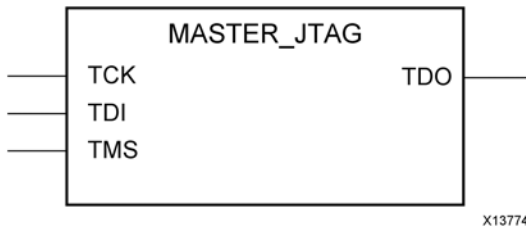
## For More Information

- See the *UltraScale Architecture Configurable Logic Block User Guide* ([UG574](#)).
- See the [UltraScale Programmable Devices Documentation](#).

# MASTER\_JTAG

Primitive: JTAG Port Access

PRIMITIVE\_GROUP: CONFIGURATION  
PRIMITIVE\_SUBGROUP: MASTER\_JTAG  
Families: UltraScale, UltraScale+



## Introduction

This component is only intended for advanced secure applications, such as any combination of AES key programming (BBRAM or EFUSE), USER EFUSE programming during runtime, and where external JTAG access is prohibited. This component is not recommended when external JTAG port access is needed (ie. Vivado Device Programmer/ILA programming or debug tools) since the component is used to override the external JTAG pins of the device, allowing full access to the JTAG port from within the device. Once instantiated the external JTAG port is completely disabled.

## Port Descriptions

Port	Direction	Width	Function
TCK	Input	1	JTAG TCK input pin
TDI	Input	1	JTAG TDI input pin
TDO	Output	1	JTAG TDO output pin
TMS	Input	1	JTAG TMS input pin

## Design Entry Method

Instantiation	Recommended
Inference	No
IP and IP Integrator Catalog	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;

-- MASTER_JTAG: JTAG Port Access
-- UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

MASTER_JTAG_inst : MASTER_JTAG
port map (
    TDO => TDO, -- 1-bit output: JTAG TDO output pin
    TCK => TCK, -- 1-bit input: JTAG TCK input pin
    TDI => TDI, -- 1-bit input: JTAG TDI input pin
    TMS => TMS -- 1-bit input: JTAG TMS input pin
);

-- End of MASTER_JTAG_inst instantiation
  
```

## Verilog Instantiation Template

```

// MASTER_JTAG: JTAG Port Access
// UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

MASTER_JTAG MASTER_JTAG_inst (
    .TDO(TDO), // 1-bit output: JTAG TDO output pin
    .TCK(TCK), // 1-bit input: JTAG TCK input pin
    .TDI(TDI), // 1-bit input: JTAG TDI input pin
    .TMS(TMS) // 1-bit input: JTAG TMS input pin
);

// End of MASTER_JTAG_inst instantiation
  
```

## For More Information

- See the *UltraScale Architecture Configuration User Guide* ([UG570](#)).
- See the [UltraScale Programmable Devices Documentation](#).



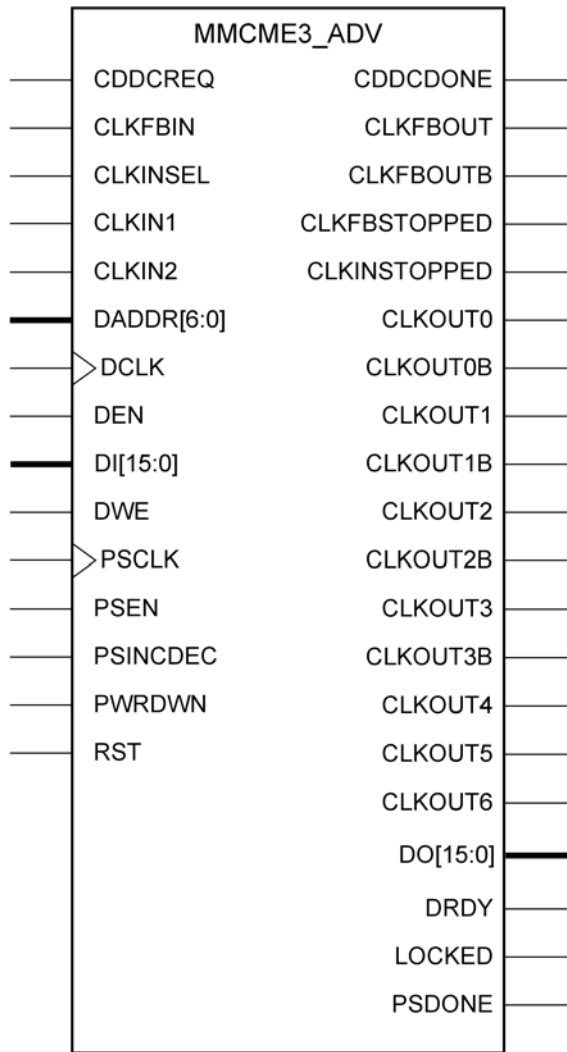
# MMCME3\_ADV

Primitive: Advanced Mixed Mode Clock Manager (MMCM)

PRIMITIVE\_GROUP: CLOCK

PRIMITIVE\_SUBGROUP: PLL

Families: UltraScale, UltraScale+



X13408

## Introduction

The MMCME3 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 MMCME3 supports dynamic phase shifting and fractional divides.

## Port Descriptions

Port	Direction	Width	Function
CDDCREQ	Input	1	Active high request signal for dynamically changing output clock divide.
<b>Clock Inputs:</b> MMCM input clock(s). Connect CLKIN2 if input clock multiplexing feature is used.			
CLKIN1	Input	1	Primary clock input.
CLKIN2	Input	1	Secondary clock input to dynamically switch the MMCM reference clock.
<b>Clock Outputs:</b> User configurable clock outputs 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. The CLKOUT#B outputs are 180 degrees out of phase with the non-B outputs.			
CLKOUT0	Output	1	CLKOUT0 output
CLKOUT0B	Output	1	User configurable clock outputs 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. The CLKOUT#B outputs are 180 degrees out of phase with the non-B outputs.
CLKOUT1	Output	1	CLKOUT1 output
CLKOUT1B	Output	1	Inverted CLKOUT2 output
CLKOUT2	Output	1	User configurable clock outputs 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. The CLKOUT#B outputs are 180 degrees out of phase with the non-B outputs.
CLKOUT2B	Output	1	Inverted CLKOUT2 output
CLKOUT3	Output	1	CLKOUT3 output
CLKOUT3B	Output	1	Inverted CLKOUT3 output
CLKOUT4	Output	1	CLKOUT4 output
CLKOUT5	Output	1	CLKOUT5 output
CLKOUT6	Output	1	CLKOUT6 output
<b>Control Ports:</b> MMCM control ports.			
CLKINSEL	Input	1	Signal controls the state of the input MUX, High = CLKIN1, Low = CLKIN2.
PWRDWN	Input	1	Powers down MMCM components, thus reducing power consumption when derived clocks are not in use for sustained periods of time. Upon release of PWRDWN, the MMCM must regain LOCK prior to use.
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 required when the input clock conditions change (e.g., frequency).
<b>DRP Ports:</b> Ports used when using the dynamic reconfigurable ports for reading and writing the configuration of the MMCM.			

Port	Direction	Width	Function
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 zeroes.
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 output (DRDY) 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.
<b>Dynamic Phase Shift Ports:</b> Ports used when using the dynamic phase shift capability of the MMCM.			
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.
<b>Feedback:</b> Required ports to form the feedback path for the MMCM phase alignment capabilities.			
CLKFBIN	Input	1	Feedback clock pin to the MMCM
CLKFBOUT	Output	1	Dedicated MMCM Feedback clock output
CLKFBOUTB	Output	1	Inverted CLKFBOUT
<b>Status Ports:</b> MMCM status ports.			
CDDCDONE	Output	1	Acknowledge signal that output clock dynamic divide is done and the output is valid.
CLKFBSTOPPED	Output	1	Status pin indicating that the feedback clock has stopped.
CLKINSTOPPED	Output	1	Status pin indicating that the input clock has stopped.
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.

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	Recommended

## Available Attributes

Attribute	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	2.000 to 64.000	5.000	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.
CLKOUT0_DIVIDE_F	3 significant digit FLOAT	1.000 to 128.000	1.000	Specifies the amount to fractional divide the associated CLKOUT clock output if a different frequency is desired. This number in combination with the CLKFBOUT_MULT_F and DIVCLK_DIVIDE values will determine the output frequency.

Attribute	Type	Allowed Values	Default	Description
COMPENSATION	STRING	"AUTO", "BUF_IN", "EXTERNAL", "INTERNAL", "ZHOLD"	"AUTO"	Clock input compensation. Should be set to ZHOLD. Defines how the MMCM feedback is configured. <ul style="list-style-type: none"> <li>"ZHOLD" - Indicates that the MMCM is configured to provide a negative hold time at the I/O registers.</li> <li>"INTERNAL" - Indicates that the MMCM is using its own internal feedback path so no delay is being compensated.</li> <li>"EXTERNAL" - Indicates that a network external to the device is being compensated.</li> <li>"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 for instance.</li> </ul>
DIVCLK_DIVIDE	DECIMAL	1 to 106	1	Specifies the division ratio for all output clocks with respect to the input clock. Effectively divides the CLKIN going into the PFD.
STARTUP_WAIT	STRING	"FALSE", "TRUE"	"FALSE"	Delays configuration DONE signal from asserting until MMCM is locked.
<b>CLKIN_PERIOD:</b> Specifies the input period in ns to the MMCM CLKIN inputs. Resolution is down to the ps. For example a value of 33.333 would indicate a 30 MHz input clock. This information is mandatory and must be supplied. CLKIN1_PERIOD relates to the input period on the CLKIN1 input while CLKIN2_PERIOD relates to the input clock period on the CLKIN2 input.				
CLKIN1_PERIOD	FLOAT(ns)	0.000 to 100.000	0.000	Specifies the input period to the CLKIN1 in ns.
CLKIN2_PERIOD	FLOAT(ns)	0.000 to 100.000	0.000	Specifies the input period to the CLKIN2 in ns.
<b>CLKOUT0_DUTY_CYCLE - CLKOUT6_DUTY_CYCLE:</b> Specifies the duty cycle of the associated CLKOUT output as a decimal number representing the percentage (i.e., 0.50 will generate a 50% duty cycle).				
CLKOUT0_DUTY_CYCLE	3 significant digit FLOAT	0.001 to 0.999	0.500	Specifies the duty cycle for CLKOUT0.
CLKOUT1_DUTY_CYCLE	3 significant digit FLOAT	0.001 to 0.999	0.500	Specifies the duty cycle for CLKOUT1.
CLKOUT2_DUTY_CYCLE	3 significant digit FLOAT	0.001 to 0.999	0.500	Specifies the duty cycle for CLKOUT2.
CLKOUT3_DUTY_CYCLE	3 significant digit FLOAT	0.001 to 0.999	0.500	Specifies the duty cycle for CLKOUT3.
CLKOUT4_DUTY_CYCLE	3 significant digit FLOAT	0.001 to 0.999	0.500	Specifies the duty cycle for CLKOUT4.

Attribute	Type	Allowed Values	Default	Description
CLKOUT5_DUTY_CYCLE	3 significant digit FLOAT	0.001 to 0.999	0.500	Specifies the duty cycle for CLKOUT5.
CLKOUT6_DUTY_CYCLE	3 significant digit FLOAT	0.001 to 0.999	0.500	Specifies the duty cycle for CLKOUT6.
<b>CLKOUT0_PHASE - CLKOUT6_PHASE:</b> 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.				
CLKOUT0_PHASE	3 significant digit FLOAT	-360.000 to 360.000	0.000	Specifies the phase offset for CLKOUT0.
CLKOUT1_PHASE	3 significant digit FLOAT	-360.000 to 360.000	0.000	Specifies the phase offset for CLKOUT1.
CLKOUT2_PHASE	3 significant digit FLOAT	-360.000 to 360.000	0.000	Specifies the phase offset for CLKOUT2.
CLKOUT3_PHASE	3 significant digit FLOAT	-360.000 to 360.000	0.000	Specifies the phase offset for CLKOUT3.
CLKOUT4_PHASE	3 significant digit FLOAT	-360.000 to 360.000	0.000	Specifies the phase offset for CLKOUT4.
CLKOUT5_PHASE	3 significant digit FLOAT	-360.000 to 360.000	0.000	Specifies the phase offset for CLKOUT5.
CLKOUT6_PHASE	3 significant digit FLOAT	-360.000 to 360.000	0.000	Specifies the phase offset for CLKOUT6.
<b>CLKOUT1_DIVIDE - CLKOUT6_DIVIDE:</b> Specifies the amount to divide the associated CLKOUT clock output if a different frequency is desired. This number in combination with the CLKFBOUT_MULT_F and DIVCLK_DIVIDE values will determine the output frequency.				
CLKOUT1_DIVIDE	DECIMAL	1 to 128	1	Specifies the amount by which to divide CLKOUT1 to create a different frequency.
CLKOUT2_DIVIDE	DECIMAL	1 to 128	1	Specifies the amount by which to divide CLKOUT2 to create a different frequency.
CLKOUT3_DIVIDE	DECIMAL	1 to 128	1	Specifies the amount by which to divide CLKOUT3 to create a different frequency.
CLKOUT4_CASCADE	STRING	"FALSE", "TRUE"	"FALSE"	Cascades the output divider (counter) CLKOUT6 into the input of the CLKOUT4 divider for an output clock divider that is greater than 128, effectively providing a total divide value of 16,384.
CLKOUT4_DIVIDE	DECIMAL	1 to 128	1	Specifies the amount by which to divide CLKOUT4 to create a different frequency.
CLKOUT5_DIVIDE	DECIMAL	1 to 128	1	Specifies the amount by which to divide CLKOUT5 to create a different frequency.
CLKOUT6_DIVIDE	DECIMAL	1 to 128	1	Specifies the amount by which to divide CLKOUT6 to create a different frequency.

Attribute	Type	Allowed Values	Default	Description
<b>Programmable Inversion Attributes:</b> Specifies whether or not to use the optional inversions on specific pins for this component to change the active polarity of the pin function. When set to 1 on a clock input pin (CLKIN# or CLKFBIN), the phase is effectively shifted 180 degrees. When set to 1 on other pins, it changes the function to behave active-Low rather than active-High. If an external inverter is specified on one of these associated pins, the Vivado Design Suite will automatically set this attribute during the opt_design stage so that additional logic is not necessary for changing the input polarity.				
IS_CLKFBIN_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the CLKFBIN pin.
IS_CLKINSEL_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the CLKINSEL pin.
IS_CLKIN1_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the CLKIN1 pin.
IS_CLKIN2_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the CLKIN2 pin.
IS_PSEN_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the PSEN pin.
IS_PSINCDEC_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the PSINCDEC pin.
IS_PWRDWN_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the PWRDWN pin.
IS_RST_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the RST pin.
<b>REF_JITTER:</b> Specifies the expected jitter on the CLKIN inputs 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_JITTER1	3 significant digit FLOAT	0.000 to 0.999	0.010	Specifies the expected jitter on the CLKIN1
REF_JITTER2	3 significant digit FLOAT	0.000 to 0.999	0.010	Specifies the expected jitter on the CLKIN2
<b>Spread Spectrum:</b> Spread Spectrum Attributes				
SS_EN	STRING	"FALSE", "TRUE"	"FALSE"	Enables the spread spectrum feature for the MMCM. Used in conjunction with SS_MODE and SS_MOD_PERIOD attributes.
SS_MOD_PERIOD	DECIMAL(ns)	4000 to 40000	10000	Specifies the spread spectrum modulation period (ns).
SS_MODE	STRING	"CENTER_HIGH", "CENTER_LOW", "DOWN_HIGH", "DOWN_LOW"	"CENTER_HIGH"	Controls the spread spectrum frequency deviation and the spread type.
<b>USE_FINE_PS:</b> Counter variable fine phase shift enable.				

Attribute	Type	Allowed Values	Default	Description
CLKFBOUT_USE_FINE_PS	STRING	"FALSE", "TRUE"	"FALSE"	Counter variable fine phase shift enable.
CLKOUT0_USE_FINE_PS	STRING	"FALSE", "TRUE"	"FALSE"	Counter variable fine phase shift enable.
CLKOUT1_USE_FINE_PS	STRING	"FALSE", "TRUE"	"FALSE"	Counter variable fine phase shift enable.
CLKOUT2_USE_FINE_PS	STRING	"FALSE", "TRUE"	"FALSE"	Counter variable fine phase shift enable.
CLKOUT3_USE_FINE_PS	STRING	"FALSE", "TRUE"	"FALSE"	Counter variable fine phase shift enable.
CLKOUT4_USE_FINE_PS	STRING	"FALSE", "TRUE"	"FALSE"	Counter variable fine phase shift enable.
CLKOUT5_USE_FINE_PS	STRING	"FALSE", "TRUE"	"FALSE"	Counter variable fine phase shift enable.
CLKOUT6_USE_FINE_PS	STRING	"FALSE", "TRUE"	"FALSE"	Counter 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;

-- MMCME3_ADV: Advanced Mixed Mode Clock Manager (MMCM)
-- UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

MMCME3_ADV_inst : MMCME3_ADV
generic map (
    BANDWIDTH => "OPTIMIZED",           -- Jitter programming (HIGH, LOW, OPTIMIZED)
    CLKFBOUT_MULT_F => 5.0,             -- Multiply value for all CLKOUT (2.000-64.000)
    CLKFBOUT_PHASE => 0.0,             -- Phase offset in degrees of CLKFB (-360.000-360.000)
    -- CLKIN_PERIOD: Input clock period in ns units, 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 (0.001-0.999).
    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_CASCADE => "FALSE",
CLKOUT4_DIVIDE => 1,
CLKOUT5_DIVIDE => 1,
CLKOUT6_DIVIDE => 1,
COMPENSATION => "AUTO",           -- AUTO, BUF_IN, EXTERNAL, INTERNAL, ZHOLD
DIVCLK_DIVIDE => 1,               -- Master division value (1-106)
-- Programmable Inversion Attributes: Specifies built-in programmable inversion on specific pins
IS_CLKFBIN_INVERTED => '0',      -- Optional inversion for CLKFBIN
IS_CLKIN1_INVERTED => '0',      -- Optional inversion for CLKIN1
IS_CLKIN2_INVERTED => '0',      -- Optional inversion for CLKIN2
IS_CLKINSEL_INVERTED => '0',    -- Optional inversion for CLKINSEL
IS_PSEN_INVERTED => '0',        -- Optional inversion for PSEN
IS_PSINCDEC_INVERTED => '0',    -- Optional inversion for PSINCDEC
IS_PWRDWN_INVERTED => '0',      -- Optional inversion for PWRDWN
IS_RST_INVERTED => '0',         -- Optional inversion for RST
-- REF_JITTER: Reference input jitter in UI (0.000-0.999).
REF_JITTER1 => 0.0,
REF_JITTER2 => 0.0,
STARTUP_WAIT => "FALSE",        -- Delays DONE until MMCM is locked (FALSE, TRUE)
-- Spread Spectrum: Spread Spectrum Attributes
SS_EN => "FALSE",               -- Enables spread spectrum (FALSE, TRUE)
SS_MODE => "CENTER_HIGH",       -- CENTER_HIGH, CENTER_LOW, DOWN_HIGH, DOWN_LOW
SS_MOD_PERIOD => 10000,         -- Spread spectrum modulation period (ns) (4000-40000)
-- 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 outputs: User configurable clock outputs
CLKOUT0 => CLKOUT0,             -- 1-bit output: CLKOUT0
CLKOUT0B => CLKOUT0B,          -- 1-bit output: Inverted CLKOUT0
CLKOUT1 => CLKOUT1,             -- 1-bit output: Primary clock
CLKOUT1B => CLKOUT1B,          -- 1-bit output: Inverted CLKOUT1
CLKOUT2 => CLKOUT2,             -- 1-bit output: CLKOUT2
CLKOUT2B => CLKOUT2B,          -- 1-bit output: Inverted CLKOUT2
CLKOUT3 => CLKOUT3,             -- 1-bit output: CLKOUT3
CLKOUT3B => CLKOUT3B,          -- 1-bit output: Inverted CLKOUT3
CLKOUT4 => CLKOUT4,             -- 1-bit output: CLKOUT4
CLKOUT5 => CLKOUT5,             -- 1-bit output: CLKOUT5
CLKOUT6 => CLKOUT6,             -- 1-bit output: CLKOUT6
-- DRP Ports outputs: Dynamic reconfiguration ports
DO => DO,                       -- 16-bit output: DRP data
DRDY => DRDY,                   -- 1-bit output: DRP ready
-- Dynamic Phase Shift Ports outputs: Ports used for dynamic phase shifting of the outputs
PSDONE => PSDONE,               -- 1-bit output: Phase shift done
-- Feedback outputs: Clock feedback ports
CLKFBOUT => CLKFBOUT,           -- 1-bit output: Feedback clock
CLKFBOUTB => CLKFBOUTB,         -- 1-bit output: Inverted CLKFBOUT
-- Status Ports outputs: MMCM status ports
CDDCDONE => CDDCDONE,           -- 1-bit output: Clock dynamic divide done
CLKFBSTOPPED => CLKFBSTOPPED,   -- 1-bit output: Feedback clock stopped
CLKINSTOPPED => CLKINSTOPPED,   -- 1-bit output: Input clock stopped
LOCKED => LOCKED,               -- 1-bit output: LOCK
CDDCREQ => CDDCREQ,             -- 1-bit input: Request to dynamic divide clock
-- Clock Inputs inputs: Clock inputs
CLKIN1 => CLKIN1,               -- 1-bit input: Primary clock
CLKIN2 => CLKIN2,               -- 1-bit input: Secondary clock
-- Control Ports inputs: MMCM control ports
CLKINSEL => CLKINSEL,           -- 1-bit input: Clock select, High=CLKIN1 Low=CLKIN2
PWRDWN => PWRDWN,               -- 1-bit input: Power-down
RST => RST,                     -- 1-bit input: Reset
-- DRP Ports inputs: Dynamic reconfiguration ports
DADDR => DADDR,                 -- 7-bit input: DRP address
DCLK => DCLK,                   -- 1-bit input: DRP clock
DEN => DEN,                     -- 1-bit input: DRP enable
DI => DI,                       -- 16-bit input: DRP data

```

```

DWE => DWE,                -- 1-bit input: DRP write enable
-- Dynamic Phase Shift Ports inputs: Ports used for dynamic phase shifting of the outputs
PCLK => PCLK,              -- 1-bit input: Phase shift clock
PSEN => PSEN,              -- 1-bit input: Phase shift enable
PSINCDEC => PSINCDEC,     -- 1-bit input: Phase shift increment/decrement
-- Feedback inputs: Clock feedback ports
CLKFBIN => CLKFBIN        -- 1-bit input: Feedback clock
);

-- End of MMCME3_ADV_inst instantiation
    
```

## Verilog Instantiation Template

```

// MMCME3_ADV: Advanced Mixed Mode Clock Manager (MMCM)
//      UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

MMCME3_ADV #(
    .BANDWIDTH("OPTIMIZED"),          // Jitter programming (HIGH, LOW, OPTIMIZED)
    .CLKFBOUT_MULT_F(5.0),            // Multiply value for all CLKOUT (2.000-64.000)
    .CLKFBOUT_PHASE(0.0),            // Phase offset in degrees of CLKFB (-360.000-360.000)
    // CLKIN_PERIOD: Input clock period in ns units, 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 (0.001-0.999).
    .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_CASCADE("FALSE"),
    .CLKOUT4_DIVIDE(1),
    .CLKOUT5_DIVIDE(1),
    .CLKOUT6_DIVIDE(1),
    .COMPENSATION("AUTO"),           // AUTO, BUF_IN, EXTERNAL, INTERNAL, ZHOLD
    .DIVCLK_DIVIDE(1),              // Master division value (1-106)
    // Programmable Inversion Attributes: Specifies built-in programmable inversion on specific pins
    .IS_CLKFBIN_INVERTED(1'b0),     // Optional inversion for CLKFBIN
    .IS_CLKIN1_INVERTED(1'b0),      // Optional inversion for CLKIN1
    .IS_CLKIN2_INVERTED(1'b0),      // Optional inversion for CLKIN2
    .IS_CLKINSEL_INVERTED(1'b0),    // Optional inversion for CLKINSEL
    .IS_PSEN_INVERTED(1'b0),        // Optional inversion for PSEN
    .IS_PSINCDEC_INVERTED(1'b0),    // Optional inversion for PSINCDEC
    .IS_PWRDWN_INVERTED(1'b0),      // Optional inversion for PWRDWN
    .IS_RST_INVERTED(1'b0),         // Optional inversion for RST
    // REF_JITTER: Reference input jitter in UI (0.000-0.999).
    .REF_JITTER1(0.0),
    .REF_JITTER2(0.0),
    .STARTUP_WAIT("FALSE"),         // Delays DONE until MMCM is locked (FALSE, TRUE)
    // Spread Spectrum: Spread Spectrum Attributes
    .SS_EN("FALSE"),                // Enables spread spectrum (FALSE, TRUE)
    .SS_MODE("CENTER_HIGH"),        // CENTER_HIGH, CENTER_LOW, DOWN_HIGH, DOWN_LOW
    .SS_MOD_PERIOD(10000),          // Spread spectrum modulation period (ns) (4000-40000)
    // 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")
    )
    MMCME3_ADV_inst (
        // Clock Outputs outputs: User configurable clock outputs
        .CLKOUT0 (CLKOUT0),           // 1-bit output: CLKOUT0
        .CLKOUT0B (CLKOUT0B),        // 1-bit output: Inverted CLKOUT0
        .CLKOUT1 (CLKOUT1),           // 1-bit output: Primary clock
        .CLKOUT1B (CLKOUT1B),        // 1-bit output: Inverted CLKOUT1
        .CLKOUT2 (CLKOUT2),           // 1-bit output: CLKOUT2
        .CLKOUT2B (CLKOUT2B),        // 1-bit output: Inverted CLKOUT2
        .CLKOUT3 (CLKOUT3),           // 1-bit output: CLKOUT3
        .CLKOUT3B (CLKOUT3B),        // 1-bit output: Inverted CLKOUT3
        .CLKOUT4 (CLKOUT4),           // 1-bit output: CLKOUT4
        .CLKOUT5 (CLKOUT5),           // 1-bit output: CLKOUT5
        .CLKOUT6 (CLKOUT6),           // 1-bit output: CLKOUT6
        // DRP Ports outputs: Dynamic reconfiguration ports
        .DO (DO),                     // 16-bit output: DRP data
        .DRDY (DRDY),                 // 1-bit output: DRP ready
        // Dynamic Phase Shift Ports outputs: Ports used for dynamic phase shifting of the outputs
        .PSDONE (PSDONE),             // 1-bit output: Phase shift done
        // Feedback outputs: Clock feedback ports
        .CLKFBOUT (CLKFBOUT),         // 1-bit output: Feedback clock
        .CLKFBOUTB (CLKFBOUTB),       // 1-bit output: Inverted CLKFBOUT
        // Status Ports outputs: MCM status ports
        .CDDCDONE (CDDCDONE),         // 1-bit output: Clock dynamic divide done
        .CLKFBSTOPPED (CLKFBSTOPPED), // 1-bit output: Feedback clock stopped
        .CLKINSTOPPED (CLKINSTOPPED), // 1-bit output: Input clock stopped
        .LOCKED (LOCKED),             // 1-bit output: LOCK
        .CDDCREQ (CDDCREQ),           // 1-bit input: Request to dynamic divide clock
        // Clock Inputs inputs: Clock inputs
        .CLKIN1 (CLKIN1),             // 1-bit input: Primary clock
        .CLKIN2 (CLKIN2),             // 1-bit input: Secondary clock
        // Control Ports inputs: MCM control ports
        .CLKINSEL (CLKINSEL),         // 1-bit input: Clock select, High=CLKIN1 Low=CLKIN2
        .PWRDWN (PWRDWN),             // 1-bit input: Power-down
        .RST (RST),                   // 1-bit input: Reset
        // DRP Ports inputs: Dynamic reconfiguration ports
        .DADDR (DADDR),               // 7-bit input: DRP address
        .DCLK (DCLK),                 // 1-bit input: DRP clock
        .DEN (DEN),                   // 1-bit input: DRP enable
        .DI (DI),                     // 16-bit input: DRP data
        .DWE (DWE),                   // 1-bit input: DRP write enable
        // Dynamic Phase Shift Ports inputs: Ports used for dynamic phase shifting of the outputs
        .PSCLK (PSCLK),               // 1-bit input: Phase shift clock
        .PSEN (PSEN),                 // 1-bit input: Phase shift enable
        .PSINCDEC (PSINCDEC),         // 1-bit input: Phase shift increment/decrement
        // Feedback inputs: Clock feedback ports
        .CLKFBIN (CLKFBIN)           // 1-bit input: Feedback clock
    );

    // End of MMCME3_ADV_inst instantiation

```

## For More Information

- See the *UltraScale Architecture Clocking Resources User Guide* ([UG572](#)).
- See the [UltraScale Programmable Devices Documentation](#).

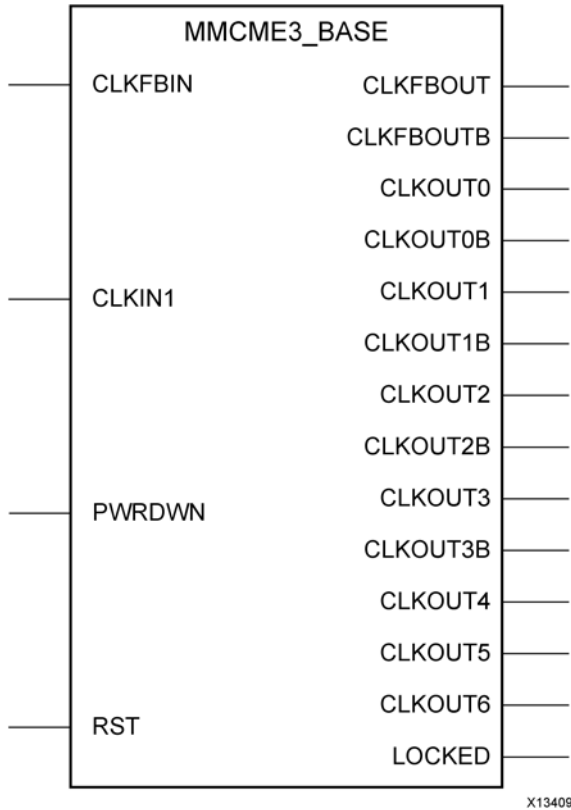
# MMCM3\_BASE

Primitive: Base Mixed Mode Clock Manager (MMCM)

PRIMITIVE\_GROUP: CLOCK

PRIMITIVE\_SUBGROUP: PLL

Families: UltraScale, UltraScale+



## Introduction

The MMCME3 is a mixed signal block designed to support frequency synthesis, clock network deskew, phase adjustment and jitter reduction. The MMCME3\_BASE supports a subset of the more common features and thus is easier to instantiate and use compared to the full features MMCME3\_ADV.

## Port Descriptions

Port	Direction	Width	Function
<b>Clock Inputs:</b> MMCM input clock.			
CLKIN1	Input	1	General clock input.
<b>Clock Outputs:</b> User configurable clock outputs 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. The CLKOUT#B outputs are 180 degrees out of phase with the non-B outputs.			
CLKOUT0	Output	1	CLKOUT0 output
CLKOUT0B	Output	1	Inverted CLKOUT0 output
CLKOUT1	Output	1	CLKOUT1 output
CLKOUT1B	Output	1	Inverted CLKOUT1 output
CLKOUT2	Output	1	CLKOUT2 output
CLKOUT2B	Output	1	Inverted CLKOUT2 output
CLKOUT3	Output	1	CLKOUT3 output
CLKOUT3B	Output	1	Inverted CLKOUT3 output
CLKOUT4	Output	1	CLKOUT4 output
CLKOUT5	Output	1	CLKOUT5 output
CLKOUT6	Output	1	CLKOUT6 output
<b>Control Ports:</b> MMCM control ports.			
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 required when the input clock conditions change (e.g., frequency).
<b>Feedback:</b> Required ports to form the feedback path for the MMCM phase alignment capabilities.			
CLKFBIN	Input	1	Feedback clock pin to the MMCM
CLKFBOUT	Output	1	Dedicated MMCM Feedback clock output
CLKFBOUTB	Output	1	Inverted CLKFBOUT output
<b>Status Ports:</b> MMCM status ports.			
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.

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	Recommended

## Available Attributes

Attribute	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	2.000 to 64.000	5.000	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)	0.000 to 100.000	0.000	Specifies the input period in ns to the MMCM CLKIN1 input. Resolution is down to the ps (3 decimal places). For example, a value of 33.333 would indicate a 30 MHz input clock. 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_F and DIVCLK_DIVIDE values will determine the output frequency.
CLKOUT4_CASCADE	STRING	"FALSE", "TRUE"	"FALSE"	Cascades the output divider (counter) CLKOUT6 into the input of the CLKOUT4 divider for an output clock divider that is greater than 128.
DIVCLK_DIVIDE	DECIMAL	1 to 106	1	Specifies the division ratio for all output clocks with respect to the input clock. Effectively divides the CLKIN going into the PFD.

Attribute	Type	Allowed Values	Default	Description
REF_JITTER1	3 significant digit FLOAT	0.000 to 0.999	0.010	Specifies 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.
STARTUP_WAIT	STRING	"FALSE", "TRUE"	"FALSE"	Delays configuration DONE signal from asserting until MMCM is locked.
<b>CLKOUT0_DUTY_CYCLE - CLKOUT6_DUTY_CYCLE:</b> Specifies the duty cycle of the associated CLKOUT output as a decimal number representing the percentage (i.e., 0.50 will generate a 50% duty cycle).				
CLKOUT0_DUTY_CYCLE	3 significant digit FLOAT	0.001 to 0.999	0.500	Specifies the duty cycle for CLKOUT0.
CLKOUT1_DUTY_CYCLE	3 significant digit FLOAT	0.001 to 0.999	0.500	Specifies the duty cycle for CLKOUT1.
CLKOUT2_DUTY_CYCLE	3 significant digit FLOAT	0.001 to 0.999	0.500	Specifies the duty cycle for CLKOUT2.
CLKOUT3_DUTY_CYCLE	3 significant digit FLOAT	0.001 to 0.999	0.500	Specifies the duty cycle for CLKOUT3.
CLKOUT4_DUTY_CYCLE	3 significant digit FLOAT	0.001 to 0.999	0.500	Specifies the duty cycle for CLKOUT4.
CLKOUT5_DUTY_CYCLE	3 significant digit FLOAT	0.001 to 0.999	0.500	Specifies the duty cycle for CLKOUT5.
CLKOUT6_DUTY_CYCLE	3 significant digit FLOAT	0.001 to 0.999	0.500	Specifies the duty cycle for CLKOUT6.
<b>CLKOUT0_PHASE - CLKOUT6_PHASE:</b> 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.				
CLKOUT0_PHASE	3 significant digit FLOAT	-360.000 to 360.000	0.000	Specifies the phase offset for CLKOUT0.
CLKOUT1_PHASE	3 significant digit FLOAT	-360.000 to 360.000	0.000	Specifies the phase offset for CLKOUT1.
CLKOUT2_PHASE	3 significant digit FLOAT	-360.000 to 360.000	0.000	Specifies the phase offset for CLKOUT2.
CLKOUT3_PHASE	3 significant digit FLOAT	-360.000 to 360.000	0.000	Specifies the phase offset for CLKOUT3.
CLKOUT4_PHASE	3 significant digit FLOAT	-360.000 to 360.000	0.000	Specifies the phase offset for CLKOUT4.
CLKOUT5_PHASE	3 significant digit FLOAT	-360.000 to 360.000	0.000	Specifies the phase offset for CLKOUT5.
CLKOUT6_PHASE	3 significant digit FLOAT	-360.000 to 360.000	0.000	Specifies the phase offset for CLKOUT6.
<b>CLKOUT1_DIVIDE - CLKOUT6_DIVIDE:</b> Specifies the amount to divide the associated CLKOUT clock output if a different frequency is desired. This number in combination with the CLKFBOUT_MULT_F and DIVCLK_DIVIDE values will determine the output frequency.				

Attribute	Type	Allowed Values	Default	Description
CLKOUT1_DIVIDE	DECIMAL	1 to 128	1	Specifies the amount by which to divide CLKOUT1 to create a different frequency.
CLKOUT2_DIVIDE	DECIMAL	1 to 128	1	Specifies the amount by which to divide CLKOUT2 to create a different frequency.
CLKOUT3_DIVIDE	DECIMAL	1 to 128	1	Specifies the amount by which to divide CLKOUT3 to create a different frequency.
CLKOUT4_DIVIDE	DECIMAL	1 to 128	1	Specifies the amount by which to divide CLKOUT4 to create a different frequency.
CLKOUT5_DIVIDE	DECIMAL	1 to 128	1	Specifies the amount by which to divide CLKOUT5 to create a different frequency.
CLKOUT6_DIVIDE	DECIMAL	1 to 128	1	Specifies the amount by which to divide CLKOUT6 to create a different frequency.
<b>Programmable Inversion Attributes:</b> Specifies whether or not to use the optional inversions on specific pins of this component to change the active polarity of the pin function. When set to 1 on a clock input pin (CLKIN1 or CLKFBIN), the phase is effectively shifted 180 degrees. When set to 1 on other pins, it changes the function to behave active-Low rather than active-High. If an external inverter is specified on one of these associated pins, the Vivado Design Suite will automatically set this attribute during the opt_design stage so that additional logic is not necessary for changing the input polarity.				
IS_CLKFBIN_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the CLKFBIN pin.
IS_CLKIN1_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the CLKIN1 pin.
IS_PWRDWN_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the PWRDWN pin.
IS_RST_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the RST pin.

## VHDL Instantiation Template

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- MMCME3_BASE: Base Mixed Mode Clock Manager (MMCM)
-- UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

MMCME3_BASE_inst : MMCME3_BASE
generic map (
    BANDWIDTH => "OPTIMIZED",    -- Jitter programming (HIGH, LOW, OPTIMIZED)
    CLKFBOUT_MULT_F => 5.0,      -- Multiply value for all CLKOUT (2.000-64.000)
    CLKFBOUT_PHASE => 0.0,      -- Phase offset in degrees of CLKFB (-360.000-360.000)
    CLKIN1_PERIOD => 0.0,       -- Input clock period in ns units, 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 (0.001-0.999).
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", -- Cascade CLKOUT4 counter with CLKOUT6 (FALSE, TRUE)
DIVCLK_DIVIDE => 1, -- Master division value (1-106)
-- Programmable Inversion Attributes: Specifies built-in programmable inversion on specific pins
IS_CLKFBIN_INVERTED => '0', -- Optional inversion for CLKFBIN
IS_CLKIN1_INVERTED => '0', -- Optional inversion for CLKIN1
IS_PWRDWN_INVERTED => '0', -- Optional inversion for PWRDWN
IS_RST_INVERTED => '0', -- Optional inversion for RST
REF_JITTER1 => 0.0, -- Reference input jitter in UI (0.000-0.999)
STARTUP_WAIT => "FALSE" -- Delays DONE until MMCM is locked (FALSE, TRUE)
)
port map (
-- Clock Outputs outputs: User configurable clock outputs
CLKOUT0 => CLKOUT0, -- 1-bit output: CLKOUT0
CLKOUT0B => CLKOUT0B, -- 1-bit output: Inverted CLKOUT0
CLKOUT1 => CLKOUT1, -- 1-bit output: CLKOUT1
CLKOUT1B => CLKOUT1B, -- 1-bit output: Inverted CLKOUT1
CLKOUT2 => CLKOUT2, -- 1-bit output: CLKOUT2
CLKOUT2B => CLKOUT2B, -- 1-bit output: Inverted CLKOUT2
CLKOUT3 => CLKOUT3, -- 1-bit output: CLKOUT3
CLKOUT3B => CLKOUT3B, -- 1-bit output: Inverted CLKOUT3
CLKOUT4 => CLKOUT4, -- 1-bit output: CLKOUT4
CLKOUT5 => CLKOUT5, -- 1-bit output: CLKOUT5
CLKOUT6 => CLKOUT6, -- 1-bit output: CLKOUT6
-- Feedback outputs: Clock feedback ports
CLKFBOUT => CLKFBOUT, -- 1-bit output: Feedback clock
CLKFBOUTB => CLKFBOUTB, -- 1-bit output: Inverted CLKFBOUT
-- Status Ports outputs: MMCM status ports
LOCKED => LOCKED, -- 1-bit output: LOCK
-- Clock Inputs inputs: Clock input
CLKIN1 => CLKIN1, -- 1-bit input: Clock
-- Control Ports inputs: MMCM control ports
PWRDWN => PWRDWN, -- 1-bit input: Power-down
RST => RST, -- 1-bit input: Reset
-- Feedback inputs: Clock feedback ports
CLKFBIN => CLKFBIN -- 1-bit input: Feedback clock
);

-- End of MMCME3_BASE_inst instantiation
    
```

## Verilog Instantiation Template

```

// MMCME3_BASE: Base Mixed Mode Clock Manager (MMCM)
// UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

MMCME3_BASE #(
    .BANDWIDTH("OPTIMIZED"), // Jitter programming (HIGH, LOW, OPTIMIZED)
    
```

```

        .CLKFBOUT_MULT F(5.0),          // Multiply value for all CLKOUT (2.000-64.000)
        .CLKFBOUT_PHASE(0.0),         // Phase offset in degrees of CLKFB (-360.000-360.000)
        .CLKIN1_PERIOD(0.0),          // Input clock period in ns units, 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 (0.001-0.999).
        .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"),     // Cascade CLKOUT4 counter with CLKOUT6 (FALSE, TRUE)
        .DIVCLK_DIVIDE(1),             // Master division value (1-106)
        // Programmable Inversion Attributes: Specifies built-in programmable inversion on specific pins
        .IS_CLKFBIN_INVERTED(1'b0),   // Optional inversion for CLKFBIN
        .IS_CLKIN1_INVERTED(1'b0),    // Optional inversion for CLKIN1
        .IS_PWRDWN_INVERTED(1'b0),    // Optional inversion for PWRDWN
        .IS_RST_INVERTED(1'b0),       // Optional inversion for RST
        .REF_JITTER1(0.0),            // Reference input jitter in UI (0.000-0.999)
        .STARTUP_WAIT("FALSE")        // Delays DONE until MMCM is locked (FALSE, TRUE)
    )
    MMCME3_BASE_inst (
        // Clock Outputs outputs: User configurable clock outputs
        .CLKOUT0(CLKOUT0),             // 1-bit output: CLKOUT0
        .CLKOUT0B(CLKOUT0B),           // 1-bit output: Inverted CLKOUT0
        .CLKOUT1(CLKOUT1),             // 1-bit output: CLKOUT1
        .CLKOUT1B(CLKOUT1B),           // 1-bit output: Inverted CLKOUT1
        .CLKOUT2(CLKOUT2),             // 1-bit output: CLKOUT2
        .CLKOUT2B(CLKOUT2B),           // 1-bit output: Inverted CLKOUT2
        .CLKOUT3(CLKOUT3),             // 1-bit output: CLKOUT3
        .CLKOUT3B(CLKOUT3B),           // 1-bit output: Inverted CLKOUT3
        .CLKOUT4(CLKOUT4),             // 1-bit output: CLKOUT4
        .CLKOUT5(CLKOUT5),             // 1-bit output: CLKOUT5
        .CLKOUT6(CLKOUT6),             // 1-bit output: CLKOUT6
        // Feedback outputs: Clock feedback ports
        .CLKFBOUT(CLKFBOUT),           // 1-bit output: Feedback clock
        .CLKFBOUTB(CLKFBOUTB),         // 1-bit output: Inverted CLKFBOUT
        // Status Ports outputs: MMCM status ports
        .LOCKED(LOCKED),               // 1-bit output: LOCK
        // Clock Inputs inputs: Clock input
        .CLKIN1(CLKIN1),               // 1-bit input: Clock
        // Control Ports inputs: MMCM control ports
        .PWRDWN(PWRDWN),               // 1-bit input: Power-down
        .RST(RST),                     // 1-bit input: Reset
        // Feedback inputs: Clock feedback ports
        .CLKFBIN(CLKFBIN)              // 1-bit input: Feedback clock
    );

// End of MMCME3_BASE_inst instantiation

```

## For More Information

- See the *UltraScale Architecture Clocking Resources User Guide* ([UG572](#)).
- See the [UltraScale Programmable Devices Documentation](#).

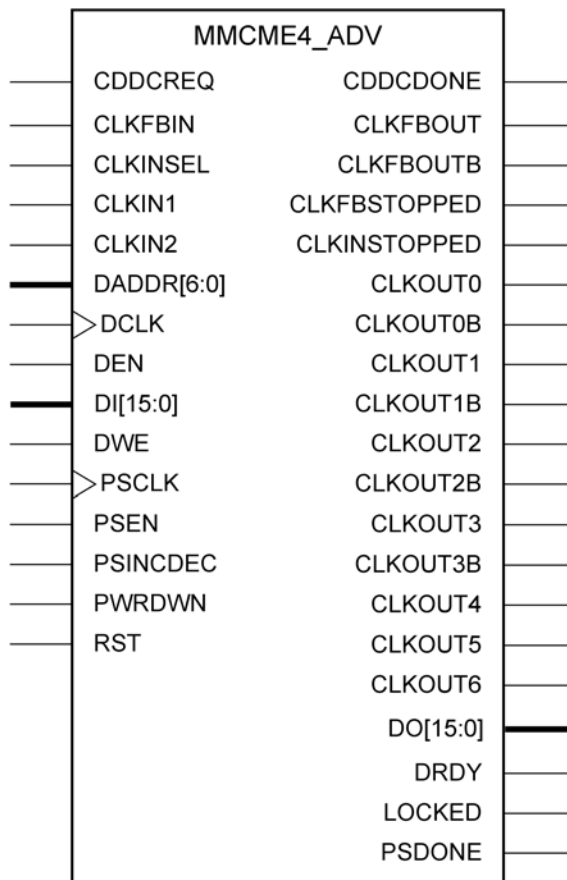
# MMCME4\_ADV

Primitive: Advanced Mixed Mode Clock Manager (MMCM)

PRIMITIVE\_GROUP: CLOCK

PRIMITIVE\_SUBGROUP: PLL

Families: UltraScale+



X15107-102615

## Introduction

The MMCME4 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 MMCME4 supports dynamic phase shifting and fractional divides.

## Port Descriptions

Port	Direction	Width	Function
CDDCDONE	Output	1	Acknowledge signal that output clock dynamic divide is done and the output is valid

Port	Direction	Width	Function
CDDCREQ	Input	1	Active high request signal for dynamically changing output clock divide
CLKFBIN	Input	1	Feedback clock pin to the MMCM
CLKFBOUT	Output	1	Dedicated MMCM Feedback clock output
CLKFBOUTB	Output	1	Inverted CLKFBOUT
CLKFBSTOPPED	Output	1	Status pin indicating that the feedback clock has stopped
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	Primary clock input
CLKIN2	Input	1	Secondary clock input to dynamically switch the MMCM reference clock
CLKOUT0	Output	1	CLKOUT0 output
CLKOUT0B	Output	1	Inverted CLKOUT0
CLKOUT1	Output	1	CLKOUT1 output
CLKOUT1B	Output	1	Inverted CLKOUT1
CLKOUT2	Output	1	CLKOUT2 output
CLKOUT2B	Output	1	Inverted CLKOUT2
CLKOUT3	Output	1	CLKOUT3 output
CLKOUT3B	Output	1	Inverted CLKOUT3
CLKOUT4	Output	1	CLKOUT4 output
CLKOUT5	Output	1	CLKOUT5 output
CLKOUT6	Output	1	CLKOUT6 output
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 zeroes
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 output (DRDY) provides the response to the DEN signal for the MMCMs dynamic reconfiguration feature

Port	Direction	Width	Function
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 MMCM components, thus reducing power consumption when derived clocks are not in use for sustained periods of time. Upon release of PWRDWN, the MMCM must regain LOCK prior to use
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 required when the input clock conditions change (e.g., frequency)

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	Recommended

## Available Attributes

Attribute	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	2.000 to 128.000	5.000	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

Attribute	Type	Allowed Values	Default	Description
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
CLKFBOUT_USE_FINE_PS	STRING	"FALSE", "TRUE"	"FALSE"	Counter variable fine phase shift enable
CLKIN1_PERIOD	FLOAT(ns)	0.000 to 100.000	0.000	Specifies the input period to the CLKIN1 in ns
CLKIN2_PERIOD	FLOAT(ns)	0.000 to 100.000	0.000	Specifies the input period to the CLKIN2 in ns
CLKOUT0_DIVIDE_F	3 significant digit FLOAT	1.000 to 128.000	1.000	Specifies the amount to fractional divide the associated CLKOUT clock output if a different frequency is desired. This number in combination with the CLKFBOUT_MULT_F and DIVCLK_DIVIDE values will determine the output frequency
CLKOUT0_DUTY_CYCLE	3 significant digit FLOAT	0.001 to 0.999	0.500	Specifies the Duty Cycle of CLKOUT0 clock output in percentage (i.e., 0.500 will generate a 50% duty cycle)
CLKOUT0_PHASE	3 significant digit FLOAT	-360.000 to 360.000	0.000	Specifies the phase offset in degrees of the CLKOUT0 output
CLKOUT0_USE_FINE_PS	STRING	"FALSE", "TRUE"	"FALSE"	Counter variable fine phase shift enable
CLKOUT1_DIVIDE	DECIMAL	1 to 128	1	Specifies the amount by which to divide CLKOUT1 to create a different frequency
CLKOUT1_DUTY_CYCLE	3 significant digit FLOAT	0.001 to 0.999	0.500	Specifies the duty cycle for CLKOUT1
CLKOUT1_PHASE	3 significant digit FLOAT	-360.000 to 360.000	0.000	Specifies the phase offset for CLKOUT1
CLKOUT1_USE_FINE_PS	STRING	"FALSE", "TRUE"	"FALSE"	Counter variable fine phase shift enable
CLKOUT2_DIVIDE	DECIMAL	1 to 128	1	Specifies the amount by which to divide CLKOUT2 to create a different frequency
CLKOUT2_DUTY_CYCLE	3 significant digit FLOAT	0.001 to 0.999	0.500	Specifies the duty cycle for CLKOUT2
CLKOUT2_PHASE	3 significant digit FLOAT	-360.000 to 360.000	0.000	Specifies the phase offset for CLKOUT2
CLKOUT2_USE_FINE_PS	STRING	"FALSE", "TRUE"	"FALSE"	Counter variable fine phase shift enable
CLKOUT3_DIVIDE	DECIMAL	1 to 128	1	Specifies the amount by which to divide CLKOUT3 to create a different frequency

Attribute	Type	Allowed Values	Default	Description
CLKOUT3_DUTY_CYCLE	3 significant digit FLOAT	0.001 to 0.999	0.500	Specifies the duty cycle for CLKOUT3
CLKOUT3_PHASE	3 significant digit FLOAT	-360.000 to 360.000	0.000	Specifies the phase offset for CLKOUT3
CLKOUT3_USE_FINE_PS	STRING	"FALSE", "TRUE"	"FALSE"	Counter variable fine phase shift enable
CLKOUT4_CASCADE	STRING	"FALSE", "TRUE"	"FALSE"	Cascades the output divider (counter) CLKOUT6 into the input of the CLKOUT4 divider for an output clock divider that is greater than 128, effectively providing a total divide value of 16,384
CLKOUT4_DIVIDE	DECIMAL	1 to 128	1	Specifies the amount by which to divide CLKOUT4 to create a different frequency
CLKOUT4_DUTY_CYCLE	3 significant digit FLOAT	0.001 to 0.999	0.500	Specifies the duty cycle for CLKOUT4
CLKOUT4_PHASE	3 significant digit FLOAT	-360.000 to 360.000	0.000	Specifies the phase offset for CLKOUT4
CLKOUT4_USE_FINE_PS	STRING	"FALSE", "TRUE"	"FALSE"	Counter variable fine phase shift enable
CLKOUT5_DIVIDE	DECIMAL	1 to 128	1	Specifies the amount by which to divide CLKOUT5 to create a different frequency
CLKOUT5_DUTY_CYCLE	3 significant digit FLOAT	0.001 to 0.999	0.500	Specifies the duty cycle for CLKOUT5
CLKOUT5_PHASE	3 significant digit FLOAT	-360.000 to 360.000	0.000	Specifies the phase offset for CLKOUT5
CLKOUT5_USE_FINE_PS	STRING	"FALSE", "TRUE"	"FALSE"	Counter variable fine phase shift enable
CLKOUT6_DIVIDE	DECIMAL	1 to 128	1	Specifies the amount by which to divide CLKOUT6 to create a different frequency
CLKOUT6_DUTY_CYCLE	3 significant digit FLOAT	0.001 to 0.999	0.500	Specifies the duty cycle for CLKOUT6
CLKOUT6_PHASE	3 significant digit FLOAT	-360.000 to 360.000	0.000	Specifies the phase offset for CLKOUT6
CLKOUT6_USE_FINE_PS	STRING	"FALSE", "TRUE"	"FALSE"	Counter variable fine phase shift enable

Attribute	Type	Allowed Values	Default	Description
COMPENSATION	STRING	"AUTO", "BUF_IN", "EXTERNAL", "INTERNAL", "ZHOLD"	"AUTO"	Clock input compensation. Should be set to ZHOLD. Defines how the MMCM feedback is configured. <ul style="list-style-type: none"> <li>"ZHOLD" - Indicates that the MMCM is configured to provide a negative hold time at the I/O registers.</li> <li>"INTERNAL" - Indicates that the MMCM is using its own internal feedback path so no delay is being compensated.</li> <li>"EXTERNAL" - Indicates that a network external to the device is being compensated.</li> <li>"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 for instance.</li> </ul>
DIVCLK_DIVIDE	DECIMAL	1 to 106	1	Specifies the division ratio for all output clocks with respect to the input clock. Effectively divides the CLKIN going into the PFD
IS_CLKFBIN_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the CLKFBIN pin
IS_CLKINSEL_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the CLKINSEL pin
IS_CLKIN1_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the CLKIN1 pin
IS_CLKIN2_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the CLKIN2 pin
IS_PSEN_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the PSEN pin
IS_PSINCDEC_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the PSINCDEC pin
IS_PWRDWN_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the PWRDWN pin
IS_RST_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the RST pin
REF_JITTER1	3 significant digit FLOAT	0.000 to 0.999	0.010	Specifies the expected jitter on the CLKIN1



Attribute	Type	Allowed Values	Default	Description
REF_JITTER2	3 significant digit FLOAT	0.000 to 0.999	0.010	Specifies the expected jitter on the CLKIN2
SS_EN	STRING	"FALSE", "TRUE"	"FALSE"	Enables the spread spectrum feature for the MMCM. Used in conjunction with SS_MODE and SS_MOD_PERIOD attributes
SS_MOD_PERIOD	DECIMAL(ns)	4000 to 40000	10000	Specifies the spread spectrum modulation period (ns)
SS_MODE	STRING	"CENTER_HIGH", "CENTER_LOW", "DOWN_HIGH", "DOWN_LOW"	"CENTER_HIGH"	Controls the spread spectrum frequency deviation and the spread type
STARTUP_WAIT	STRING	"FALSE", "TRUE"	"FALSE"	Delays configuration DONE signal from asserting until MMCM is locked

## VHDL Instantiation Template

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- MMCME4_ADV: Advanced Mixed Mode Clock Manager (MMCM)
--           UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

MMCME4_ADV_inst : MMCME4_ADV
generic map (
    BANDWIDTH => "OPTIMIZED",           -- Jitter programming
    CLKFBOUT_MULT_F => 5.0,             -- Multiply value for all CLKOUT
    CLKFBOUT_PHASE => 0.0,             -- Phase offset in degrees of CLKFB
    CLKFBOUT_USE_FINE_PS => "FALSE",    -- Fine phase shift enable (TRUE/FALSE)
    CLKIN1_PERIOD => 0.0,              -- Input clock period in ns to ps resolution (i.e. 33.333 is 30 MHz).
    CLKIN2_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
    CLKOUT0_DUTY_CYCLE => 0.5,         -- Duty cycle for CLKOUT0
    CLKOUT0_PHASE => 0.0,              -- Phase offset for CLKOUT0
    CLKOUT0_USE_FINE_PS => "FALSE",    -- Fine phase shift enable (TRUE/FALSE)
    CLKOUT1_DIVIDE => 1,                -- Divide amount for CLKOUT (1-128)
    CLKOUT1_DUTY_CYCLE => 0.5,         -- Duty cycle for CLKOUT outputs (0.001-0.999).
    CLKOUT1_PHASE => 0.0,              -- Phase offset for CLKOUT outputs (-360.000-360.000).
    CLKOUT1_USE_FINE_PS => "FALSE",    -- Fine phase shift enable (TRUE/FALSE)
    CLKOUT2_DIVIDE => 1,                -- Divide amount for CLKOUT (1-128)
    CLKOUT2_DUTY_CYCLE => 0.5,         -- Duty cycle for CLKOUT outputs (0.001-0.999).
    CLKOUT2_PHASE => 0.0,              -- Phase offset for CLKOUT outputs (-360.000-360.000).
    CLKOUT2_USE_FINE_PS => "FALSE",    -- Fine phase shift enable (TRUE/FALSE)
    CLKOUT3_DIVIDE => 1,                -- Divide amount for CLKOUT (1-128)
    CLKOUT3_DUTY_CYCLE => 0.5,         -- Duty cycle for CLKOUT outputs (0.001-0.999).
    CLKOUT3_PHASE => 0.0,              -- Phase offset for CLKOUT outputs (-360.000-360.000).
    CLKOUT3_USE_FINE_PS => "FALSE",    -- Fine phase shift enable (TRUE/FALSE)
    CLKOUT4_CASCADE => "FALSE",        -- Divide amount for CLKOUT (1-128)
    CLKOUT4_DIVIDE => 1,                -- Divide amount for CLKOUT (1-128)
    CLKOUT4_DUTY_CYCLE => 0.5,         -- Duty cycle for CLKOUT outputs (0.001-0.999).
    CLKOUT4_PHASE => 0.0,              -- Phase offset for CLKOUT outputs (-360.000-360.000).
    CLKOUT4_USE_FINE_PS => "FALSE",    -- Fine phase shift enable (TRUE/FALSE)
    CLKOUT5_DIVIDE => 1,                -- Divide amount for CLKOUT (1-128)
    CLKOUT5_DUTY_CYCLE => 0.5,         -- Duty cycle for CLKOUT outputs (0.001-0.999).
    CLKOUT5_PHASE => 0.0,              -- Phase offset for CLKOUT outputs (-360.000-360.000).
    CLKOUT5_USE_FINE_PS => "FALSE",    -- Fine phase shift enable (TRUE/FALSE)

```

```

CLKOUT6_DIVIDE => 1,           -- Divide amount for CLKOUT (1-128)
CLKOUT6_DUTY_CYCLE => 0.5,    -- Duty cycle for CLKOUT outputs (0.001-0.999).
CLKOUT6_PHASE => 0.0,         -- Phase offset for CLKOUT outputs (-360.000-360.000).
CLKOUT6_USE_FINE_PS => "FALSE", -- Fine phase shift enable (TRUE/FALSE)
COMPENSATION => "AUTO",       -- Clock input compensation
DIVCLK_DIVIDE => 1,           -- Master division value
IS_CLKFBIN_INVERTED => '0',   -- Optional inversion for CLKFBIN
IS_CLKIN1_INVERTED => '0',    -- Optional inversion for CLKIN1
IS_CLKIN2_INVERTED => '0',    -- Optional inversion for CLKIN2
IS_CLKINSEL_INVERTED => '0',  -- Optional inversion for CLKINSEL
IS_PSEN_INVERTED => '0',      -- Optional inversion for PSEN
IS_PSINCDEC_INVERTED => '0',  -- Optional inversion for PSINCDEC
IS_PWRDWN_INVERTED => '0',    -- Optional inversion for PWRDWN
IS_RST_INVERTED => '0',       -- Optional inversion for RST
REF_JITTER1 => 0.0,           -- Reference input jitter in UI (0.000-0.999).
REF_JITTER2 => 0.0,           -- Reference input jitter in UI (0.000-0.999).
SS_EN => "FALSE",            -- Enables spread spectrum
SS_MODE => "CENTER_HIGH",     -- Spread spectrum frequency deviation and the spread type
SS_MOD_PERIOD => 10000,       -- Spread spectrum modulation period (ns)
STARTUP_WAIT => "FALSE"      -- Delays DONE until MMCM is locked
)
port map (
  CDDCDONE => CDDCDONE,        -- 1-bit output: Clock dynamic divide done
  CLKFBOUT => CLKFBOUT,        -- 1-bit output: Feedback clock
  CLKFBOUTB => CLKFBOUTB,      -- 1-bit output: Inverted CLKFBOUT
  CLKFBSTOPPED => CLKFBSTOPPED, -- 1-bit output: Feedback clock stopped
  CLKINSTOPPED => CLKINSTOPPED, -- 1-bit output: Input clock stopped
  CLKOUT0 => CLKOUT0,          -- 1-bit output: CLKOUT0
  CLKOUT0B => CLKOUT0B,        -- 1-bit output: Inverted CLKOUT0
  CLKOUT1 => CLKOUT1,          -- 1-bit output: CLKOUT1
  CLKOUT1B => CLKOUT1B,        -- 1-bit output: Inverted CLKOUT1
  CLKOUT2 => CLKOUT2,          -- 1-bit output: CLKOUT2
  CLKOUT2B => CLKOUT2B,        -- 1-bit output: Inverted CLKOUT2
  CLKOUT3 => CLKOUT3,          -- 1-bit output: CLKOUT3
  CLKOUT3B => CLKOUT3B,        -- 1-bit output: Inverted CLKOUT3
  CLKOUT4 => CLKOUT4,          -- 1-bit output: CLKOUT4
  CLKOUT5 => CLKOUT5,          -- 1-bit output: CLKOUT5
  CLKOUT6 => CLKOUT6,          -- 1-bit output: CLKOUT6
  DO => DO,                    -- 16-bit output: DRP data output
  DRDY => DRDY,                -- 1-bit output: DRP ready
  LOCKED => LOCKED,            -- 1-bit output: LOCK
  PSDONE => PSDONE,            -- 1-bit output: Phase shift done
  CDDCREQ => CDDCREQ,          -- 1-bit input: Request to dynamic divide clock
  CLKFBIN => CLKFBIN,          -- 1-bit input: Feedback clock
  CLKIN1 => CLKIN1,            -- 1-bit input: Primary clock
  CLKIN2 => CLKIN2,            -- 1-bit input: Secondary clock
  CLKINSEL => CLKINSEL,        -- 1-bit input: Clock select, High=CLKIN1 Low=CLKIN2
  DADDR => DADDR,              -- 7-bit input: DRP address
  DCLK => DCLK,                 -- 1-bit input: DRP clock
  DEN => DEN,                   -- 1-bit input: DRP enable
  DI => DI,                      -- 16-bit input: DRP data input
  DWE => DWE,                    -- 1-bit input: DRP write enable
  PSCLK => PSCLK,               -- 1-bit input: Phase shift clock
  PSEN => PSEN,                  -- 1-bit input: Phase shift enable
  PSINCDEC => PSINCDEC,         -- 1-bit input: Phase shift increment/decrement
  PWRDWN => PWRDWN,             -- 1-bit input: Power-down
  RST => RST                      -- 1-bit input: Reset
);
-- End of MMCME4_ADV_inst instantiation

```

## Verilog Instantiation Template

```

// MMCME4_ADV: Advanced Mixed Mode Clock Manager (MMCM)
//      UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

MMCME4_ADV #(
  .BANDWIDTH("OPTIMIZED"), // Jitter programming
  .CLKFBOUT_MULT_F(5.0),   // Multiply value for all CLKOUT

```

```

.CLKFBOUT_PHASE(0.0), // Phase offset in degrees of CLKFB
.CLKFBOUT_USE_FINE_PS("FALSE"), // Fine phase shift enable (TRUE/FALSE)
.CLKIN1_PERIOD(0.0), // Input clock period in ns to ps resolution (i.e. 33.333 is 30 MHz).
.CLKIN2_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
.CLKOUT0_DUTY_CYCLE(0.5), // Duty cycle for CLKOUT0
.CLKOUT0_PHASE(0.0), // Phase offset for CLKOUT0
.CLKOUT0_USE_FINE_PS("FALSE"), // Fine phase shift enable (TRUE/FALSE)
.CLKOUT1_DIVIDE(1), // Divide amount for CLKOUT (1-128)
.CLKOUT1_DUTY_CYCLE(0.5), // Duty cycle for CLKOUT outputs (0.001-0.999).
.CLKOUT1_PHASE(0.0), // Phase offset for CLKOUT outputs (-360.000-360.000).
.CLKOUT1_USE_FINE_PS("FALSE"), // Fine phase shift enable (TRUE/FALSE)
.CLKOUT2_DIVIDE(1), // Divide amount for CLKOUT (1-128)
.CLKOUT2_DUTY_CYCLE(0.5), // Duty cycle for CLKOUT outputs (0.001-0.999).
.CLKOUT2_PHASE(0.0), // Phase offset for CLKOUT outputs (-360.000-360.000).
.CLKOUT2_USE_FINE_PS("FALSE"), // Fine phase shift enable (TRUE/FALSE)
.CLKOUT3_DIVIDE(1), // Divide amount for CLKOUT (1-128)
.CLKOUT3_DUTY_CYCLE(0.5), // Duty cycle for CLKOUT outputs (0.001-0.999).
.CLKOUT3_PHASE(0.0), // Phase offset for CLKOUT outputs (-360.000-360.000).
.CLKOUT3_USE_FINE_PS("FALSE"), // Fine phase shift enable (TRUE/FALSE)
.CLKOUT4_CASCADE("FALSE"), // Divide amount for CLKOUT (1-128)
.CLKOUT4_DIVIDE(1), // Divide amount for CLKOUT (1-128)
.CLKOUT4_DUTY_CYCLE(0.5), // Duty cycle for CLKOUT outputs (0.001-0.999).
.CLKOUT4_PHASE(0.0), // Phase offset for CLKOUT outputs (-360.000-360.000).
.CLKOUT4_USE_FINE_PS("FALSE"), // Fine phase shift enable (TRUE/FALSE)
.CLKOUT5_DIVIDE(1), // Divide amount for CLKOUT (1-128)
.CLKOUT5_DUTY_CYCLE(0.5), // Duty cycle for CLKOUT outputs (0.001-0.999).
.CLKOUT5_PHASE(0.0), // Phase offset for CLKOUT outputs (-360.000-360.000).
.CLKOUT5_USE_FINE_PS("FALSE"), // Fine phase shift enable (TRUE/FALSE)
.CLKOUT6_DIVIDE(1), // Divide amount for CLKOUT (1-128)
.CLKOUT6_DUTY_CYCLE(0.5), // Duty cycle for CLKOUT outputs (0.001-0.999).
.CLKOUT6_PHASE(0.0), // Phase offset for CLKOUT outputs (-360.000-360.000).
.CLKOUT6_USE_FINE_PS("FALSE"), // Fine phase shift enable (TRUE/FALSE)
.COMPENSATION("AUTO"), // Clock input compensation
.DIVCLK_DIVIDE(1), // Master division value
.IS_CLKFBIN_INVERTED(1'b0), // Optional inversion for CLKFBIN
.IS_CLKIN1_INVERTED(1'b0), // Optional inversion for CLKIN1
.IS_CLKIN2_INVERTED(1'b0), // Optional inversion for CLKIN2
.IS_CLKINSEL_INVERTED(1'b0), // Optional inversion for CLKINSEL
.IS_PSEN_INVERTED(1'b0), // Optional inversion for PSEN
.IS_PSINCDEC_INVERTED(1'b0), // Optional inversion for PSINCDEC
.IS_PWRDWN_INVERTED(1'b0), // Optional inversion for PWRDWN
.IS_RST_INVERTED(1'b0), // Optional inversion for RST
.REF_JITTER1(0.0), // Reference input jitter in UI (0.000-0.999).
.REF_JITTER2(0.0), // Reference input jitter in UI (0.000-0.999).
.SS_EN("FALSE"), // Enables spread spectrum
.SS_MODE("CENTER_HIGH"), // Spread spectrum frequency deviation and the spread type
.SS_MOD_PERIOD(10000), // Spread spectrum modulation period (ns)
.STARTUP_WAIT("FALSE") // Delays DONE until MMCM is locked
)
MMCME4_ADV_inst (
.CDDCDONE(CDDCDONE), // 1-bit output: Clock dynamic divide done
.CLKFBOUT(CLKFBOUT), // 1-bit output: Feedback clock
.CLKFBOUTB(CLKFBOUTB), // 1-bit output: Inverted CLKFBOUT
.CLKFBSTOPPED(CLKFBSTOPPED), // 1-bit output: Feedback clock stopped
.CLKINSTOPPED(CLKINSTOPPED), // 1-bit output: Input clock stopped
.CLKOUT0(CLKOUT0), // 1-bit output: CLKOUT0
.CLKOUT0B(CLKOUT0B), // 1-bit output: Inverted CLKOUT0
.CLKOUT1(CLKOUT1), // 1-bit output: CLKOUT1
.CLKOUT1B(CLKOUT1B), // 1-bit output: Inverted CLKOUT1
.CLKOUT2(CLKOUT2), // 1-bit output: CLKOUT2
.CLKOUT2B(CLKOUT2B), // 1-bit output: Inverted CLKOUT2
.CLKOUT3(CLKOUT3), // 1-bit output: CLKOUT3
.CLKOUT3B(CLKOUT3B), // 1-bit output: Inverted CLKOUT3
.CLKOUT4(CLKOUT4), // 1-bit output: CLKOUT4
.CLKOUT5(CLKOUT5), // 1-bit output: CLKOUT5
.CLKOUT6(CLKOUT6), // 1-bit output: CLKOUT6
.DO(DO), // 16-bit output: DRP data output
.DRDY(DRDY), // 1-bit output: DRP ready
.LOCKED(LOCKED), // 1-bit output: LOCK
.PSDONE(PSDONE), // 1-bit output: Phase shift done
.CDDCREQ(CDDCREQ), // 1-bit input: Request to dynamic divide clock

```

```

        .CLKFBIN(CLKFBIN),           // 1-bit input: Feedback clock
        .CLKIN1(CLKIN1),           // 1-bit input: Primary clock
        .CLKIN2(CLKIN2),           // 1-bit input: Secondary clock
        .CLKINSEL(CLKINSEL),       // 1-bit input: Clock select, High=CLKIN1 Low=CLKIN2
        .DADDR(DADDR),            // 7-bit input: DRP address
        .DCLK(DCLK),              // 1-bit input: DRP clock
        .DEN(DEN),                // 1-bit input: DRP enable
        .DI(DI),                  // 16-bit input: DRP data input
        .DWE(DWE),                // 1-bit input: DRP write enable
        .PSCLK(PSCLK),            // 1-bit input: Phase shift clock
        .PSEN(PSEN),              // 1-bit input: Phase shift enable
        .PSINCDEC(PSINCDEC),       // 1-bit input: Phase shift increment/decrement
        .PWRDWN(PWRDWN),          // 1-bit input: Power-down
        .RST(RST)                  // 1-bit input: Reset
    );

// End of MMCME4_ADV_inst instantiation
    
```

## For More Information

- See the *UltraScale Architecture Clocking Resources User Guide* ([UG572](#)).
- See the [UltraScale Programmable Devices Documentation](#).

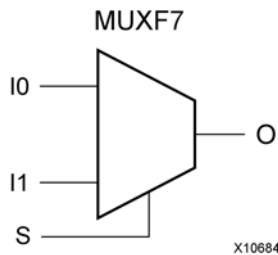
## MUXF7

Primitive: CLB MUX to connect two LUT6's Together

PRIMITIVE\_GROUP: CLB

PRIMITIVE\_SUBGROUP: MUXF

Families: UltraScale, UltraScale+



### Introduction

This design element is a two input multiplexer which, in combination with two LUT6 elements will let you create any 7-input logic function, an 8-to-1 multiplexer, or other logic functions up to 13-bits wide all within a single CLB. Outputs of the LUT6 elements are connected to the I0 and I1 inputs of the MUXF7. The S input is driven from any 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
I0	Input	1	MUX data input. Connects to the output of a LUT6 located in the same CLB.
I1	Input	1	MUX data input. Connects to the output of a LUT6 located in the same CLB.
O	Output	1	Data output from the CLB MUX.
S	Input	1	Selects between the I0 (S=0) or I1 (S=1) inputs to the MUX.

## Design Entry Method

Instantiation	Yes
Inference	Recommended
IP and IP Integrator Catalog	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 connect two LUT6's Together
-- UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

MUXF7_inst : MUXF7
port map (
    O => O, -- 1-bit output: Output of MUX
    I0 => I0, -- 1-bit input: Connect to LUT6 output
    I1 => I1, -- 1-bit input: Connect to LUT6 output
    S => S -- 1-bit input: Input select to MUX
);

-- End of MUXF7_inst instantiation
```

## Verilog Instantiation Template

```
// MUXF7: CLB MUX to connect two LUT6's Together
// UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

MUXF7 MUXF7_inst (
    .O(O), // 1-bit output: Output of MUX
    .I0(I0), // 1-bit input: Connect to LUT6 output
    .I1(I1), // 1-bit input: Connect to LUT6 output
    .S(S) // 1-bit input: Input select to MUX
);

// End of MUXF7_inst instantiation
```

## For More Information

- See the *UltraScale Architecture Configurable Logic Block User Guide* ([UG574](#)).
- See the [UltraScale Programmable Devices Documentation](#).

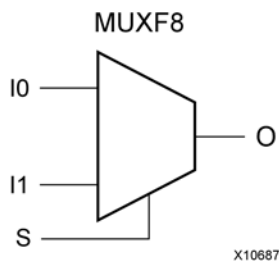
## MUXF8

Primitive: CLB MUX to connect two MUXF7's Together

PRIMITIVE\_GROUP: CLB

PRIMITIVE\_SUBGROUP: MUXF

Families: UltraScale, UltraScale+



### Introduction

This design element is a two input multiplexer which, in combination with two MUXF7 and four LUT6 elements will let you create any 8-input logic function, an 16-to-1 multiplexer, or other logic functions up to 27-bits wide all within a single CLB. Outputs of the MUXF7 elements are connected to the I0 and I1 inputs of the MUXF8. The S input is driven from any 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
I0	Input	1	MUX data input. Connects to the output of a MUXF7 located in the same CLB.
I1	Input	1	MUX data input. Connects to the output of a MUXF7 located in the same CLB.
O	Output	1	Data output from the CLB MUX.
S	Input	1	Selects between the I0 (S=0) or I1 (S=1) inputs to the MUX.

## Design Entry Method

Instantiation	Yes
Inference	Recommended
IP and IP Integrator Catalog	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 connect two MUXF7's Together
--      UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

MUXF8_inst : MUXF8
port map (
    O => O,    -- 1-bit output: Output of MUX
    I0 => I0,  -- 1-bit input: Connect to MUXF7 output
    I1 => I1,  -- 1-bit input: Connect to MUXF7 output
    S => S     -- 1-bit input: Input select to MUX
);

-- End of MUXF8_inst instantiation
    
```

## Verilog Instantiation Template

```

// MUXF8: CLB MUX to connect two MUXF7's Together
//      UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

MUXF8 MUXF8_inst (
    .O(O),    // 1-bit output: Output of MUX
    .I0(I0), // 1-bit input: Connect to MUXF7 output
    .I1(I1), // 1-bit input: Connect to MUXF7 output
    .S(S)    // 1-bit input: Input select to MUX
);

// End of MUXF8_inst instantiation
    
```

## For More Information

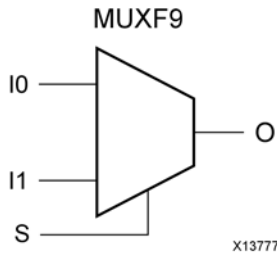
- See the *UltraScale Architecture Configurable Logic Block User Guide* ([UG574](#)).
- See the [UltraScale Programmable Devices Documentation](#).



# MUXF9

Primitive: CLB MUX to connect two MUXF8's Together

PRIMITIVE\_GROUP: CLB  
 PRIMITIVE\_SUBGROUP: MUXF  
 Families: UltraScale, UltraScale+



## Introduction

This design element is a two input multiplexer which, in combination with two MUXF8s, four MUXF7s and eight LUT6 elements will let you create any 9-input logic function, a 32-to-1 multiplexer, or other logic functions up to 55-bits wide all within a single CLB. Outputs of the MUXF8 elements are connected to the I0 and I1 inputs of the MUXF9. The S input is driven from any 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
I0	Input	1	MUX data input. Connects to the output of a MUXF8 located in the same CLB.
I1	Input	1	MUX data input. Connects to the output of a MUXF8 located in the same CLB.
O	Output	1	Data output from the CLB MUX.
S	Input	1	Selects between the I0 (S=0) or I1 (S=1) inputs to the MUX.

## Design Entry Method

Instantiation	Yes
Inference	Recommended
IP and IP Integrator Catalog	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;

-- MUXF9: CLB MUX to connect two MUXF8's Together
--      UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

MUXF9_inst : MUXF9
port map (
    O => O,    -- 1-bit output: Output of MUX
    I0 => I0,  -- 1-bit input: Connect to MUXF8 output
    I1 => I1,  -- 1-bit input: Connect to MUXF8 output
    S => S     -- 1-bit input: Input select to MUX
);

-- End of MUXF9_inst instantiation
    
```

## Verilog Instantiation Template

```

// MUXF9: CLB MUX to connect two MUXF8's Together
//      UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

MUXF9 MUXF9_inst (
    .O(O),    // 1-bit output: Output of MUX
    .I0(I0), // 1-bit input: Connect to MUXF8 output
    .I1(I1), // 1-bit input: Connect to MUXF8 output
    .S(S)    // 1-bit input: Input select to MUX
);

// End of MUXF9_inst instantiation
    
```

## For More Information

- See the *UltraScale Architecture Configurable Logic Block User Guide* ([UG574](#)).
- See the [UltraScale Programmable Devices Documentation](#).

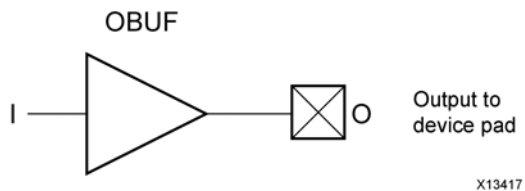
## OBUF

### Primitive: Output Buffer

PRIMITIVE\_GROUP: I/O

PRIMITIVE\_SUBGROUP: OUTPUT\_BUFFER

Families: UltraScale, UltraScale+



## Introduction

An output buffer (OBUF) must be used to drive signals from the device to external output pads.

I/O attributes that do not impact the logic function of the component, such as IOSTANDARD, DRIVE, and SLEW, should be supplied to the top-level port via an appropriate property. For details on applying such properties to the associated port, see UG912: Vivado Design Suite Properties Reference Guide.

## Port Descriptions

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

## Design Entry Method

Instantiation	Yes
Inference	Recommended
IP and IP Integrator Catalog	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;

-- OBUF: Output Buffer
--      UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

OBUF_inst : OBUF
port map (
    O => O, -- 1-bit output: Buffer output (connect directly to top-level port)

```

```
    I => I -- 1-bit input: Buffer input
);
-- End of OBUF_inst instantiation
```

## Verilog Instantiation Template

```
// OBUF: Output Buffer
//      UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

OBUF OBUF_inst (
    .O(O), // 1-bit output: Buffer output (connect directly to top-level port)
    .I(I) // 1-bit input: Buffer input
);
// End of OBUF_inst instantiation
```

## For More Information

- See the *UltraScale Architecture SelectIO Resources User Guide* ([UG571](#)).
- See the [UltraScale Programmable Devices Documentation](#).

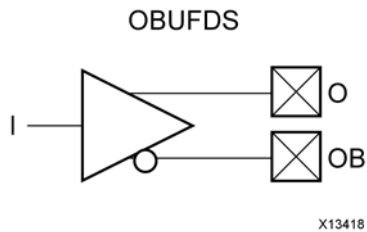
## OBUFDS

Primitive: Differential Output Buffer

PRIMITIVE\_GROUP: I/O

PRIMITIVE\_SUBGROUP: OUTPUT\_BUFFER

Families: UltraScale, UltraScale+



### Introduction

The OBUFDS is a differential output buffer primitive.

I/O attributes that do not impact the logic function of the component, such as IOSTANDARD, DRIVE, and SLEW, should be supplied to the top-level port via an appropriate property. For details on applying such properties to the associated port, see UG912: Vivado Design Suite Properties Reference Guide.

### Logic Table

Inputs	Outputs	
I	O	OB
0	0	1
1	1	0

### Port Descriptions

Port	Direction	Width	Function
I	Input	1	Buffer input
O	Output	1	Diff_p output. Connect directly to a top-level port in the design.
OB	Output	1	Diff_n output. Connect directly to a top-level port in the design.

### Design Entry Method

Instantiation	Recommended
Inference	No
IP and IP Integrator Catalog	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;

-- OBUFDS: Differential Output Buffer
-- UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

OBUFDS_inst : OBUFDS
port map (
    O => O, -- 1-bit output: Diff_p output (connect directly to top-level port)
    OB => OB, -- 1-bit output: Diff_n output (connect directly to top-level port)
    I => I -- 1-bit input: Buffer input
);

-- End of OBUFDS_inst instantiation
    
```

## Verilog Instantiation Template

```

// OBUFDS: Differential Output Buffer
// UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

OBUFDS OBUFDS_inst (
    .O(O), // 1-bit output: Diff_p output (connect directly to top-level port)
    .OB(OB), // 1-bit output: Diff_n output (connect directly to top-level port)
    .I(I) // 1-bit input: Buffer input
);

// End of OBUFDS_inst instantiation
    
```

## For More Information

- See the *UltraScale Architecture SelectIO Resources User Guide* ([UG571](#)).
- See the [UltraScale Programmable Devices Documentation](#).

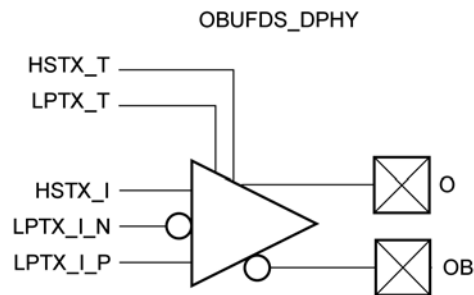
## OBUFDS\_DPHY

Primitive: Differential Output Buffer with MIPI support

PRIMITIVE\_GROUP: I/O

PRIMITIVE\_SUBGROUP: OUTPUT\_BUFFER

Families: UltraScale+



X15114-101115

### Introduction

Differential Output Buffer with MIPI support

### Port Descriptions

Port	Direction	Width	Function
HSTX_I	Input	1	Data input (HS TX)
HSTX_T	Input	1	Tristate Control input (HS TX)
LPTX_I_N	Input	1	Data input (LP TX) (Master-N)
LPTX_I_P	Input	1	Data input (LP TX) (Master-P)
LPTX_T	Input	1	Tristate Control input (LP TX)
O	Output	1	Diff_P Data output
OB	Output	1	Diff_N Data output

### Design Entry Method

Instantiation	Recommended
Inference	No
IP and IP Integrator Catalog	No

### Available Attributes

Attribute	Type	Allowed Values	Default	Description
IOSTANDARD	STRING	String	"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_DPHY: Differential Output Buffer with MIPI support
-- UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

OBUFDS_DPHY_inst : OBUFDS_DPHY
generic map (
  IOSTANDARD => "DEFAULT" -- I/O standard
)
port map (
  O => O, -- 1-bit output: Diff_P Data output
  OB => OB, -- 1-bit output: Diff_N Data output
  HSTX_I => HSTX_I, -- 1-bit input: Data input (HS TX)
  HSTX_T => HSTX_T, -- 1-bit input: Tristate Control input (HS TX)
  LPTX_I_N => LPTX_I_N, -- 1-bit input: Data input (LP TX) (Master-N)
  LPTX_I_P => LPTX_I_P, -- 1-bit input: Data input (LP TX) (Master-P)
  LPTX_T => LPTX_T -- 1-bit input: Tristate Control input (LP TX)
);

-- End of OBUFDS_DPHY_inst instantiation
  
```

## Verilog Instantiation Template

```

// OBUFDS_DPHY: Differential Output Buffer with MIPI support
// UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

OBUFDS_DPHY #(
  .IOSTANDARD("DEFAULT") // I/O standard
)
OBUFDS_DPHY_inst (
  .O(O), // 1-bit output: Diff_P Data output
  .OB(OB), // 1-bit output: Diff_N Data output
  .HSTX_I(HSTX_I), // 1-bit input: Data input (HS TX)
  .HSTX_T(HSTX_T), // 1-bit input: Tristate Control input (HS TX)
  .LPTX_I_N(LPTX_I_N), // 1-bit input: Data input (LP TX) (Master-N)
  .LPTX_I_P(LPTX_I_P), // 1-bit input: Data input (LP TX) (Master-P)
  .LPTX_T(LPTX_T) // 1-bit input: Tristate Control input (LP TX)
);

// End of OBUFDS_DPHY_inst instantiation
  
```

## For More Information

- See the *UltraScale Architecture SelectIO Resources User Guide* ([UG571](#)).
- See the [UltraScale Programmable Devices Documentation](#).



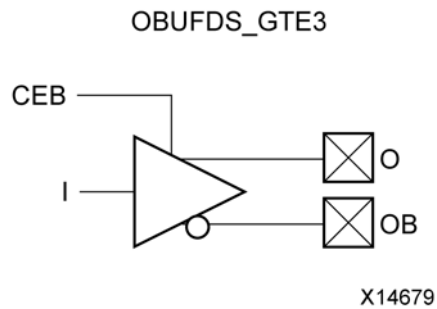
## OBUFDS\_GTE3

Primitive: Gigabit Transceiver Buffer

PRIMITIVE\_GROUP: ADVANCED

PRIMITIVE\_SUBGROUP: GT

Families: UltraScale



### Introduction

OBUFDS\_GTE3 is the gigabit transceiver output pad buffer component in UltraScale devices. The REFCLK signal should be routed to the dedicated reference clock output pins on the serial transceiver, and the user design should instantiate the OBUFDS\_GTE3 primitive in the user design. See the Transceivers User Guide for more information on PCB layout requirements, including reference clock requirements.

### Port Descriptions

Port	Direction	Width	Function
CEB	Input	1	Refer to Transceiver User Guide
I	Input	1	Refer to Transceiver User Guide
O	Output	1	Refer to Transceiver User Guide
OB	Output	1	Refer to Transceiver User Guide

### Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	Recommended

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
REFCLK_EN_TX_PATH	BINARY	1'b0 to 1'b1	1'b1	Refer to Transceiver User Guide
REFCLK_ICNTL_TX	BINARY	5'b00000 to 5'b11111	5'b00000	Refer to Transceiver User Guide

## VHDL 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_GTE3: Gigabit Transceiver Buffer
-- UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

OBUFDS_GTE3_inst : OBUFDS_GTE3
generic map (
    REFCLK_EN_TX_PATH => '1',    -- Refer to Transceiver User Guide
    REFCLK_ICNTL_TX => "00000"  -- Refer to Transceiver User Guide
)
port map (
    O => O,        -- 1-bit output: Refer to Transceiver User Guide
    OB => OB,     -- 1-bit output: Refer to Transceiver User Guide
    CEB => CEB,   -- 1-bit input: Refer to Transceiver User Guide
    I => I        -- 1-bit input: Refer to Transceiver User Guide
);

-- End of OBUFDS_GTE3_inst instantiation
```

## Verilog Instantiation Template

```
// OBUFDS_GTE3: Gigabit Transceiver Buffer
// UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

OBUFDS_GTE3 #(
    .REFCLK_EN_TX_PATH(1'b1),    // Refer to Transceiver User Guide
    .REFCLK_ICNTL_TX(5'b00000)  // Refer to Transceiver User Guide
)
OBUFDS_GTE3_inst (
    .O(O),        // 1-bit output: Refer to Transceiver User Guide
    .OB(OB),     // 1-bit output: Refer to Transceiver User Guide
    .CEB(CEB),  // 1-bit input: Refer to Transceiver User Guide
    .I(I)       // 1-bit input: Refer to Transceiver User Guide
);

// End of OBUFDS_GTE3_inst instantiation
```

## For More Information

- See the *UltraScale Architecture GTH Transceivers User Guide* ([UG576](#))
- See the *UltraScale Architecture GTY Transceivers Advance Specification User Guide* ([UG578](#))
- See the [UltraScale Programmable Devices Documentation](#).

# OBUFDS\_GTE3\_ADV

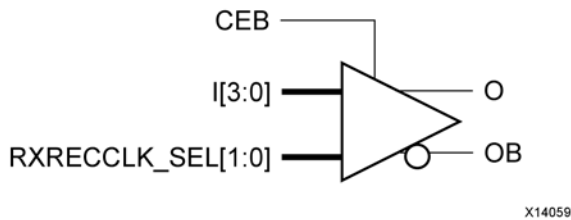
Primitive: Gigabit Transceiver Buffer

PRIMITIVE\_GROUP: ADVANCED

PRIMITIVE\_SUBGROUP: GT

Families: UltraScale

OBUFDS\_GTE3\_ADV



## Introduction

OBUFDS\_GTE3\_ADV is the gigabit transceiver output pad buffer component in UltraScale devices. The REFCLK signal should be routed to the dedicated reference clock output pins on the serial transceiver, and the user design should instantiate the OBUFDS\_GTE3 primitive in the user design. See the Transceivers User Guide for more information on PCB layout requirements, including reference clock requirements.

## Port Descriptions

Port	Direction	Width	Function
CEB	Input	1	Refer to Transceiver User Guide
I<3:0>	Input	4	Refer to Transceiver User Guide
O	Output	1	Refer to Transceiver User Guide
OB	Output	1	Refer to Transceiver User Guide
RXRECCLK_SEL<1:0>	Input	2	Refer to Transceiver User Guide

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	Recommended

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
REFCLK_EN_TX_PATH	BINARY	1'b0 to 1'b1	1'b1	Refer to Transceiver User Guide
REFCLK_ICNTL_TX	BINARY	5'b00000 to 5'b11111	5'b00000	Refer to Transceiver User Guide

## VHDL 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_GTE3_ADV: Gigabit Transceiver Buffer
--                               UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

OBUFDS_GTE3_ADV_inst : OBUFDS_GTE3_ADV
generic map (
    REFCLK_EN_TX_PATH => '1',    -- Refer to Transceiver User Guide
    REFCLK_ICNTL_TX => "00000"  -- Refer to Transceiver User Guide
)
port map (
    O => O,                      -- 1-bit output: Refer to Transceiver User Guide
    OB => OB,                    -- 1-bit output: Refer to Transceiver User Guide
    CEB => CEB,                  -- 1-bit input: Refer to Transceiver User Guide
    I => I,                      -- 4-bit input: Refer to Transceiver User Guide
    RXRECCLK_SEL => RXRECCLK_SEL -- 2-bit input: Refer to Transceiver User Guide
);

-- End of OBUFDS_GTE3_ADV_inst instantiation
    
```

## Verilog Instantiation Template

```

// OBUFDS_GTE3_ADV: Gigabit Transceiver Buffer
//                               UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

OBUFDS_GTE3_ADV #(
    .REFCLK_EN_TX_PATH(1'b1),    // Refer to Transceiver User Guide
    .REFCLK_ICNTL_TX(5'b00000)  // Refer to Transceiver User Guide
)
OBUFDS_GTE3_ADV_inst (
    .O(O),                      // 1-bit output: Refer to Transceiver User Guide
    .OB(OB),                    // 1-bit output: Refer to Transceiver User Guide
    .CEB(CEB),                  // 1-bit input: Refer to Transceiver User Guide
    .I(I),                      // 4-bit input: Refer to Transceiver User Guide
    .RXRECCLK_SEL(RXRECCLK_SEL) // 2-bit input: Refer to Transceiver User Guide
);

// End of OBUFDS_GTE3_ADV_inst instantiation
    
```

## For More Information

- See the *UltraScale Architecture GTH Transceivers User Guide* ([UG576](#))
- See the *UltraScale Architecture GTY Transceivers Advance Specification User Guide* ([UG578](#))
- See the [UltraScale Programmable Devices Documentation](#).

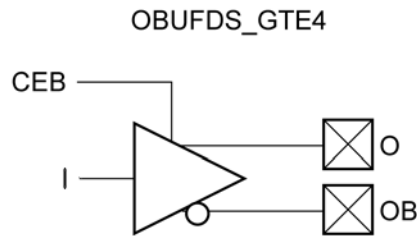
## OBUFDS\_GTE4

Primitive: Gigabit Transceiver Buffer

PRIMITIVE\_GROUP: ADVANCED

PRIMITIVE\_SUBGROUP: GT

Families: UltraScale+



X15106-101115

### Introduction

OBUFDS\_GTE4 is the gigabit transceiver output pad buffer component in UltraScale+ devices. The REFCLK signal should be routed to the dedicated reference clock output pins on the serial transceiver, and the user design should instantiate the OBUFDS\_GTE4 primitive in the user design. See the Transceivers User Guide for more information on PCB layout requirements, including reference clock requirements.

### Port Descriptions

Port	Direction	Width	Function
CEB	Input	1	Refer to Transceiver User Guide
I	Input	1	Refer to Transceiver User Guide
O	Output	1	Refer to Transceiver User Guide
OB	Output	1	Refer to Transceiver User Guide

### Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	Recommended

### Available Attributes

Attribute	Type	Allowed Values	Default	Description
REFCLK_EN_TX_PATH	BINARY	1'b0 to 1'b1	1'b1	Refer to Transceiver User Guide
REFCLK_ICNTL_TX	BINARY	5'b00000 to 5'b11111	5'b00000	Refer to Transceiver User Guide

## VHDL 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_GTE4: Gigabit Transceiver Buffer
--           UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

OBUFDS_GTE4_inst : OBUFDS_GTE4
generic map (
  REFCLK_EN_TX_PATH => '1',    -- Refer to Transceiver User Guide
  REFCLK_ICNTL_TX => "00000"  -- Refer to Transceiver User Guide
)
port map (
  O => O,    -- 1-bit output: Refer to Transceiver User Guide
  OB => OB,  -- 1-bit output: Refer to Transceiver User Guide
  CEB => CEB, -- 1-bit input: Refer to Transceiver User Guide
  I => I     -- 1-bit input: Refer to Transceiver User Guide
);

-- End of OBUFDS_GTE4_inst instantiation
  
```

## Verilog Instantiation Template

```

// OBUFDS_GTE4: Gigabit Transceiver Buffer
//           UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

OBUFDS_GTE4 #(
  .REFCLK_EN_TX_PATH(1'b1),    // Refer to Transceiver User Guide
  .REFCLK_ICNTL_TX(5'b00000)  // Refer to Transceiver User Guide
)
OBUFDS_GTE4_inst (
  .O(O),    // 1-bit output: Refer to Transceiver User Guide
  .OB(OB),  // 1-bit output: Refer to Transceiver User Guide
  .CEB(CEB), // 1-bit input: Refer to Transceiver User Guide
  .I(I)     // 1-bit input: Refer to Transceiver User Guide
);

// End of OBUFDS_GTE4_inst instantiation
  
```

## For More Information

- See the *UltraScale Architecture GTH Transceivers User Guide* ([UG576](#))
- See the *UltraScale Architecture GTY Transceivers Advance Specification User Guide* ([UG578](#))
- See the [UltraScale Programmable Devices Documentation](#).

# OBUFDS\_GTE4\_ADV

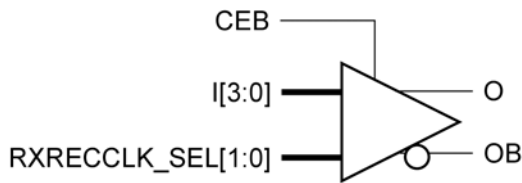
Primitive: Gigabit Transceiver Buffer

PRIMITIVE\_GROUP: ADVANCED

PRIMITIVE\_SUBGROUP: GT

Families: UltraScale+

OBUFDS\_GTE4\_ADV



X15109-101115

## Introduction

OBUFDS\_GTE4\_ADV is the gigabit transceiver output pad buffer component in UltraScale+ devices. The REFCLK signal should be routed to the dedicated reference clock output pins on the serial transceiver, and the user design should instantiate the OBUFDS\_GTE4 primitive in the user design. See the Transceivers User Guide for more information on PCB layout requirements, including reference clock requirements.

## Port Descriptions

Port	Direction	Width	Function
CEB	Input	1	Refer to Transceiver User Guide
I<3:0>	Input	4	Refer to Transceiver User Guide
O	Output	1	Refer to Transceiver User Guide
OB	Output	1	Refer to Transceiver User Guide
RXRECCLK_SEL<1:0>	Input	2	Refer to Transceiver User Guide

## Design Entry Method

Instantiation	No
Inference	Yes
IP and IP Integrator Catalog	Recommended

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
REFCLK_EN_TX_PATH	BINARY	1'b0 to 1'b1	1'b1	Refer to Transceiver User Guide
REFCLK_ICNTL_TX	BINARY	5'b00000 to 5'b11111	5'b00000	Refer to Transceiver User Guide

## VHDL 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_GTE4_ADV: Gigabit Transceiver Buffer
--                               UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

OBUFDS_GTE4_ADV_inst : OBUFDS_GTE4_ADV
generic map (
    REFCLK_EN_TX_PATH => '1',    -- Refer to Transceiver User Guide
    REFCLK_ICNTL_TX => "00000"  -- Refer to Transceiver User Guide
)
port map (
    O => O,                      -- 1-bit output: Refer to Transceiver User Guide
    OB => OB,                    -- 1-bit output: Refer to Transceiver User Guide
    CEB => CEB,                 -- 1-bit input: Refer to Transceiver User Guide
    I => I,                     -- 4-bit input: Refer to Transceiver User Guide
    RXRECCLK_SEL => RXRECCLK_SEL -- 2-bit input: Refer to Transceiver User Guide
);

-- End of OBUFDS_GTE4_ADV_inst instantiation
    
```

## Verilog Instantiation Template

```

// OBUFDS_GTE4_ADV: Gigabit Transceiver Buffer
//                               UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

OBUFDS_GTE4_ADV #(
    .REFCLK_EN_TX_PATH(1'b1),    // Refer to Transceiver User Guide
    .REFCLK_ICNTL_TX(5'b00000)  // Refer to Transceiver User Guide
)
OBUFDS_GTE4_ADV_inst (
    .O(O),                      // 1-bit output: Refer to Transceiver User Guide
    .OB(OB),                   // 1-bit output: Refer to Transceiver User Guide
    .CEB(CEB),                 // 1-bit input: Refer to Transceiver User Guide
    .I(I),                     // 4-bit input: Refer to Transceiver User Guide
    .RXRECCLK_SEL(RXRECCLK_SEL) // 2-bit input: Refer to Transceiver User Guide
);

// End of OBUFDS_GTE4_ADV_inst instantiation
    
```

## For More Information

- See the *UltraScale Architecture GTH Transceivers User Guide* ([UG576](#))
- See the *UltraScale Architecture GTY Transceivers Advance Specification User Guide* ([UG578](#))
- See the [UltraScale Programmable Devices Documentation](#).



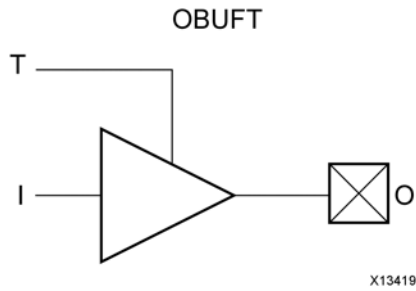
# OBUFT

Primitive: 3-State Output Buffer

PRIMITIVE\_GROUP: I/O

PRIMITIVE\_SUBGROUP: OUTPUT\_BUFFER

Families: UltraScale, UltraScale+



## Introduction

The generic 3-state output buffer OBUFT typically implements 3-state outputs or bidirectional I/O.

I/O attributes that do not impact the logic function of the component, such as IOSTANDARD, DRIVE, and SLEW, should be supplied to the top-level port via an appropriate property. For details on applying such properties to the associated port, see UG912: Vivado Design Suite Properties Reference Guide.

## Logic Table

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

## Port Descriptions

Port	Direction	Width	Function
I	Input	1	Input of OBUF. Connect to the logic driving the output port.
O	Output	1	Output of OBUF to be connected directly to top-level output port.
T	Input	1	3-state enable input

## Design Entry Method

Instantiation	Yes
Inference	Recommended
IP and IP Integrator Catalog	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;

-- OBUFT: 3-State Output Buffer
--      UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

OBUFT_inst : OBUFT
port map (
    O => O, -- 1-bit output: Buffer output (connect directly to top-level port)
    I => I, -- 1-bit input: Buffer input
    T => T  -- 1-bit input: 3-state enable input
);

-- End of OBUFT_inst instantiation
    
```

## Verilog Instantiation Template

```

// OBUFT: 3-State Output Buffer
//      UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

OBUFT OBUFT_inst (
    .O(O), // 1-bit output: Buffer output (connect directly to top-level port)
    .I(I), // 1-bit input: Buffer input
    .T(T)  // 1-bit input: 3-state enable input
);

// End of OBUFT_inst instantiation
    
```

## For More Information

- See the *UltraScale Architecture SelectIO Resources User Guide* ([UG571](#)).
- See the [UltraScale Programmable Devices Documentation](#).

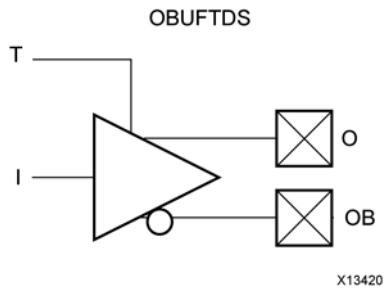
## OBUFTDS

Primitive: Differential 3-state Output Buffer

PRIMITIVE\_GROUP: I/O

PRIMITIVE\_SUBGROUP: OUTPUT\_BUFFER

Families: UltraScale, UltraScale+



### Introduction

The OBUFTDS is a differential 3-state output buffer primitive.

I/O attributes that do not impact the logic function of the component, such as IOSTANDARD and SLEW, should be supplied to the top-level port via an appropriate property. For details on applying such properties to the associated port, see UG912: Vivado Design Suite Properties Reference Guide.

### 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
I	Input	1	Buffer input
O	Output	1	Diff_p output. Connect directly to a top-level port in the design.
OB	Output	1	Diff_n output. Connect directly to a top-level port in the design.
T	Input	1	3-state enable input

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	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;

-- OBUFTDS: Differential 3-state Output Buffer
--           UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

OBUFTDS_inst : OBUFTDS
port map (
    O => O,    -- 1-bit output: Diff_p output (connect directly to top-level port)
    OB => OB,  -- 1-bit output: Diff_n output (connect directly to top-level port)
    I => I,    -- 1-bit input: Buffer input
    T => T     -- 1-bit input: 3-state enable input
);

-- End of OBUFTDS_inst instantiation
    
```

## Verilog Instantiation Template

```

// OBUFTDS: Differential 3-state Output Buffer
//           UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

OBUFTDS OBUFTDS_inst (
    .O(O),    // 1-bit output: Diff_p output (connect directly to top-level port)
    .OB(OB),  // 1-bit output: Diff_n output (connect directly to top-level port)
    .I(I),    // 1-bit input: Buffer input
    .T(T)     // 1-bit input: 3-state enable input
);

// End of OBUFTDS_inst instantiation
    
```

## For More Information

- See the *UltraScale Architecture SelectIO Resources User Guide* ([UG571](#)).
- See the [UltraScale Programmable Devices Documentation](#).

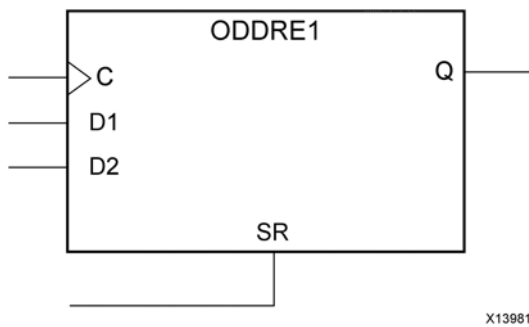
# ODDRE1

Primitive: Dedicated Dual Data Rate (DDR) Output Register

PRIMITIVE\_GROUP: REGISTER

PRIMITIVE\_SUBGROUP: DDR

Families: UltraScale, UltraScale+



## Introduction

In component mode, the ODDRE1 in UltraScale devices is a dedicated output register for use in transmitting dual data rate (DDR) signals from FPGA devices. The ODDRE1 interface with the device fabric is limited to the same clock edges. This feature allows designers to avoid additional timing complexities and CLB usage.

## Port Descriptions

Port	Direction	Width	Function
C	Input	1	High-speed clock input
D1	Input	1	Parallel data input 1
D2	Input	1	Parallel data input 2
Q	Output	1	Data output to IOB
SR	Input	1	Active High Asynchronous Reset

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	Yes

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
IS_C_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether the clock C pin is active-high or active-low.
IS_D1_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Optional inversion for D1
IS_D2_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Optional inversion for D2
SRVAL	BINARY	1'b0, 1'b1	1'b0	Initializes the ODDRE1 Flip-Flops to the specified 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;

-- ODDRE1: Dedicated Dual Data Rate (DDR) Output Register
--      UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

ODDRE1_inst : ODDRE1
generic map (
    IS_C_INVERTED => '0', -- Optional inversion for C
    IS_D1_INVERTED => '0', -- Optional inversion for D1
    IS_D2_INVERTED => '0', -- Optional inversion for D2
    SRVAL => '0'         -- Initializes the ODDRE1 Flip-Flops to the specified value ('0', '1')
)
port map (
    Q => Q,    -- 1-bit output: Data output to IOB
    C => C,    -- 1-bit input: High-speed clock input
    D1 => D1,  -- 1-bit input: Parallel data input 1
    D2 => D2,  -- 1-bit input: Parallel data input 2
    SR => SR   -- 1-bit input: Active High Async Reset
);

-- End of ODDRE1_inst instantiation
    
```

## Verilog Instantiation Template

```

// ODDRE1: Dedicated Dual Data Rate (DDR) Output Register
//      UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

ODDRE1 #(
    .IS_C_INVERTED(1'b0), // Optional inversion for C
    .IS_D1_INVERTED(1'b0), // Optional inversion for D1
    .IS_D2_INVERTED(1'b0), // Optional inversion for D2
    .SRVAL(1'b0)         // Initializes the ODDRE1 Flip-Flops to the specified value (1'b0, 1'b1)
)
ODDRE1_inst (
    .Q(Q), // 1-bit output: Data output to IOB
    .C(C), // 1-bit input: High-speed clock input
    .D1(D1), // 1-bit input: Parallel data input 1
    .D2(D2), // 1-bit input: Parallel data input 2
    .SR(SR) // 1-bit input: Active High Async Reset
);

// End of ODDRE1_inst instantiation
    
```

## For More Information

- See the *UltraScale Architecture SelectIO Resources User Guide* ([UG571](#)).
- See the [UltraScale Programmable Devices Documentation](#).

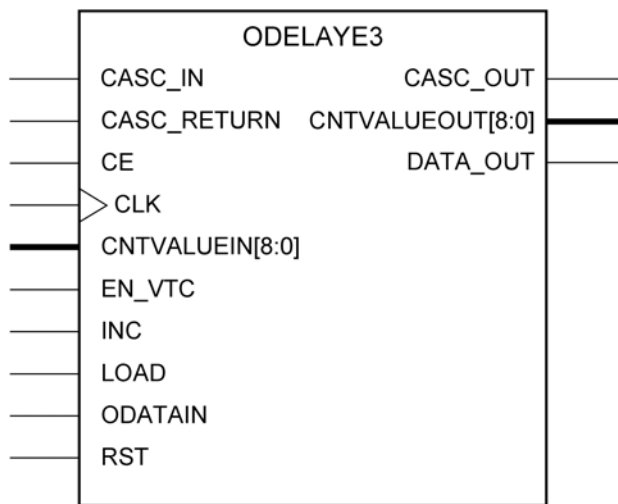
# ODELAYE3

Primitive: Output Fixed or Variable Delay Element

PRIMITIVE\_GROUP: I/O

PRIMITIVE\_SUBGROUP: DELAY

Families: UltraScale, UltraScale+



X13421

## Introduction

In component mode, I/O blocks contain a programmable delay element called ODELAYE3. The ODELAYE3 can be connected to an output register/OSERDESE3 or driven directly by device logic. The ODELAYE3 is a 512-tap delay element with a calibrated delay. The ODELAYE3 allows signals to be delayed on an individual basis.

## Port Descriptions

Port	Direction	Width	Function
CASC_IN	Input	1	Cascade delay input from slave IDELAY CASCADE_OUT
CASC_OUT	Output	1	Cascade delay output to IDELAY input cascade
CASC_RETURN	Input	1	Cascade delay returning from slave IDELAY DATAOUT
CE	Input	1	Active high enable increment/decrement function
CLK	Input	1	Clock input
CNTVALUEIN <8:0>	Input	9	Counter value from device logic for dynamically loadable tap value input
CNTVALUEOUT <8:0>	Output	9	The CNTVALUEOUT pins are used for reporting the dynamically switching value of the delay element. CNTVALUEOUT is only available when ODELAYE3 is in "VARIABLE" or "VAR_LOAD" mode.
DATAOUT	Output	1	Delayed data from ODATAIN input port
EN_VTC	Input	1	Keep delay constant over VT



Port	Direction	Width	Function
INC	Input	1	Increment/Decrement tap delay input
LOAD	Input	1	Loads the ODELAY primitive to the pre-programmed value in VARIABLE mode. In VAR_LOAD mode, it loads the value of CNTVALUEIN.
ODATAIN	Input	1	Data input for ODELAYE3 from OSERDES or programmable logic
RST	Input	1	Asynchronous Reset to the DELAY_VALUE, active level based on IS_RST_INVERTED

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	Yes

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
CASCADE	STRING	"NONE", "MASTER", "SLAVE_END", "SLAVE_MIDDLE"	"NONE"	Set the location of the ODELAYE3 when it is used in a cascaded configuration <ul style="list-style-type: none"> <li>"NONE" : Delay line is not cascaded</li> <li>"MASTER" : Delay line is cascaded with another delay line</li> <li>"SLAVE_MIDDLE" : Delay line is cascaded from adjacent delay line and also cascades to another delay line</li> <li>"SLAVE_END" : Delay line is the last cascaded delay line</li> </ul>
DELAY_FORMAT	STRING	"TIME", "COUNT"	"TIME"	Sets the units of DELAY_VALUE of the ODELAYE3. Use TIME when DELAY_TYPE is FIXED. Use COUNT when DELAY_TYPE is VARIABLE or VAR_LOAD. <ul style="list-style-type: none"> <li>"TIME" : ODELAYE3 DELAY_VALUE is specified in ps</li> <li>"COUNT" : ODELAYE3 DELAY_VALUE is specified in taps</li> </ul>

Attribute	Type	Allowed Values	Default	Description
DELAY_TYPE	STRING	"FIXED", "VAR_LOAD", "VARIABLE"	"FIXED"	Sets the type of tap delay line. <ul style="list-style-type: none"> <li>"FIXED" - Sets a static delay value</li> <li>"VARIABLE" - Dynamically adjust (increment/decrement) delay value</li> <li>"VAR_LOAD" - Dynamically loads tap values</li> </ul>
DELAY_VALUE	DECIMAL	0 to 1250	0	Specifies the fixed delay in ps in "FIXED" mode or the initial starting number of taps in "VARIABLE" mode or "VAR_LOAD" mode (input path).
IS_CLK_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether the CLK pin is active-high or active-low.
IS_RST_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether the RST pin is active-high or active-low.
REFCLK_FREQUENCY	1 significant digit FLOAT	200.0 to 2667.0	300.0	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 performance.
SIM_DEVICE	STRING	"ULTRASCALE", "ULTRASCALE_PLUS_ES1"	"ULTRASCALE"	Set the device version
UPDATE_MODE	STRING	"ASYNC", "MANUAL", "SYNC"	"ASYNC"	Determines when updates to the delay will take effect <ul style="list-style-type: none"> <li>"ASYNC": Updates are increments or decrements to the delay value independent of the data being received</li> <li>"SYNC": Updates require that ODATAIN transitions to synchronously update the delay with the ODATAIN edges</li> <li>"MANUAL": Updates take effect when both LD and CE are asserted after the LD and CNTVALUEIN signals are used to load the new CNTVALUE</li> </ul>

## VHDL Instantiation Template

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- ODELAYE3: Output Fixed or Variable Delay Element
--      UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

ODELAYE3_inst : ODELAYE3
generic map (
    CASCADE => "NONE",           -- Cascade setting (MASTER, NONE, SLAVE_END, SLAVE_MIDDLE)
    DELAY_FORMAT => "TIME",      -- (COUNT, TIME)
    DELAY_TYPE => "FIXED",      -- Set the type of tap delay line (FIXED, VARIABLE, VAR_LOAD)
    DELAY_VALUE => 0,           -- Output delay tap setting
    IS_CLK_INVERTED => '0',     -- Optional inversion for CLK
    IS_RST_INVERTED => '0',     -- Optional inversion for RST
    REFCLK_FREQUENCY => 300.0,  -- IDELAYCTRL clock input frequency in MHz (200.0-2667.0).
    SIM_DEVICE => "ULTRASCALE", -- Set the device version (ULTRASCALE, ULTRASCALE_PLUS_ES1)
    UPDATE_MODE => "ASYNC"     -- Determines when updates to the delay will take effect (ASYNC, MANUAL,
                                -- SYNC)
)
port map (
    CASC_OUT => CASC_OUT,      -- 1-bit output: Cascade delay output to IDELAY input cascade
    CNTVALUEOUT => CNTVALUEOUT, -- 9-bit output: Counter value output
    DATAOUT => DATAOUT,     -- 1-bit output: Delayed data from ODATAIN input port
    CASC_IN => CASC_IN,       -- 1-bit input: Cascade delay input from slave IDELAY CASCADE_OUT
    CASC_RETURN => CASC_RETURN, -- 1-bit input: Cascade delay returning from slave IDELAY DATAOUT
    CE => CE,                 -- 1-bit input: Active high enable increment/decrement input
    CLK => CLK,               -- 1-bit input: Clock input
    CNTVALUEIN => CNTVALUEIN, -- 9-bit input: Counter value input
    EN_VTC => EN_VTC,        -- 1-bit input: Keep delay constant over VT
    INC => INC,               -- 1-bit input: Increment/Decrement tap delay input
    LOAD => LOAD,             -- 1-bit input: Load DELAY_VALUE input
    ODATAIN => ODATAIN,      -- 1-bit input: Data input
    RST => RST                 -- 1-bit input: Asynchronous Reset to the DELAY_VALUE
);

-- End of ODELAYE3_inst instantiation
    
```

## Verilog Instantiation Template

```
// ODELAYE3: Output Fixed or Variable Delay Element
// UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

ODELAYE3 #(
    .CASCADE("NONE"),           // Cascade setting (MASTER, NONE, SLAVE_END, SLAVE_MIDDLE)
    .DELAY_FORMAT("TIME"),     // (COUNT, TIME)
    .DELAY_TYPE("FIXED"),     // Set the type of tap delay line (FIXED, VARIABLE, VAR_LOAD)
    .DELAY_VALUE(0),          // Output delay tap setting
    .IS_CLK_INVERTED(1'b0),    // Optional inversion for CLK
    .IS_RST_INVERTED(1'b0),    // Optional inversion for RST
    .REFCLK_FREQUENCY(300.0),  // IDELAYCTRL clock input frequency in MHz (200.0-2667.0).
    .SIM_DEVICE("ULTRASCALE"), // Set the device version (ULTRASCALE, ULTRASCALE_PLUS_ES1)
    .UPDATE_MODE("ASYNC")     // Determines when updates to the delay will take effect (ASYNC, MANUAL, SYNC)
)
ODELAYE3_inst (
    .CASC_OUT(CASC_OUT),       // 1-bit output: Cascade delay output to IDELAY input cascade
    .CNTVALUEOUT(CNTVALUEOUT), // 9-bit output: Counter value output
    .DATAOUT(DATAOUT),        // 1-bit output: Delayed data from ODATAIN input port
    .CASC_IN(CASC_IN),        // 1-bit input: Cascade delay input from slave IDELAY CASCADE_OUT
    .CASC_RETURN(CASC_RETURN), // 1-bit input: Cascade delay returning from slave IDELAY DATAOUT
    .CE(CE),                  // 1-bit input: Active high enable increment/decrement input
    .CLK(CLK),                // 1-bit input: Clock input
    .CNTVALUEIN(CNTVALUEIN),  // 9-bit input: Counter value input
    .EN_VTC(EN_VTC),          // 1-bit input: Keep delay constant over VT
    .INC(INC),                 // 1-bit input: Increment/Decrement tap delay input
    .LOAD(LOAD),               // 1-bit input: Load DELAY_VALUE input
    .ODATAIN(ODATAIN),        // 1-bit input: Data input
    .RST(RST)                  // 1-bit input: Asynchronous Reset to the DELAY_VALUE
);

// End of ODELAYE3_inst instantiation
```

## For More Information

- See the *UltraScale Architecture SelectIO Resources User Guide* ([UG571](#)).
- See the [UltraScale Programmable Devices Documentation](#).

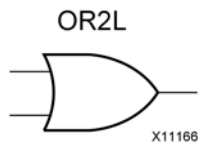
## OR2L

Primitive: Two input OR gate implemented in place of a CLB Latch

PRIMITIVE\_GROUP: CLB

PRIMITIVE\_SUBGROUP: LATCH

Families: UltraScale, UltraScale+



### Introduction

This element allows the specification of a configurable CLB latch to take the function of a two input OR gate. The use of this element can reduce logic levels and increase logic density of the part by trading off register/latch resources for logic. Xilinx suggests caution when using this component as it can affect register packing and density within a CLB.

### Logic Table

Inputs		Outputs
DI	SRI	O
0	0	0
0	1	1
1	0	1
1	1	1

### Port Descriptions

Port	Direction	Width	Function
DI	Input	1	Active-High input that is generally connected to sourcing LUT located in the same CLB.
O	Output	1	Output of the OR gate.
SRI	Input	1	Input that is generally source from outside of the CLB. The attribute IS_SRI_INVERTED determines the active polarity of this signal.  <b>NOTE:</b> To allow more than one AND2B1L or OR2L to be packed into a half CLB, a common signal must be connected to this input.

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
IS_SRI_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion for the SRI pin of this component.

## VHDL Instantiation Template

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- OR2L: Two input OR gate implemented in place of a CLB Latch
--      UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

OR2L_inst : OR2L
generic map (
    IS_SRI_INVERTED => '0' -- Optional inversion for SRI
)
port map (
    O => O,      -- 1-bit output: OR gate output
    DI => DI,    -- 1-bit input: Data input connected to LUT logic
    SRI => SRI   -- 1-bit input: External CLB data
);

-- End of OR2L_inst instantiation
    
```

## Verilog Instantiation Template

```

// OR2L: Two input OR gate implemented in place of a CLB Latch
//      UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

OR2L #(
    .IS_SRI_INVERTED(1'b0) // Optional inversion for SRI
)
OR2L_inst (
    .O(O),      // 1-bit output: OR gate output
    .DI(DI),    // 1-bit input: Data input connected to LUT logic
    .SRI(SRI)  // 1-bit input: External CLB data
);

// End of OR2L_inst instantiation
    
```

## For More Information

- See the *UltraScale Architecture Configurable Logic Block User Guide* ([UG574](#)).
- See the [UltraScale Programmable Devices Documentation](#).

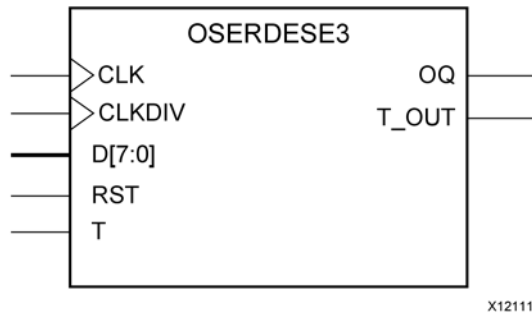
## OSERDESE3

Primitive: Output SERIAL/DESerializer

PRIMITIVE\_GROUP: I/O

PRIMITIVE\_SUBGROUP: SERDES

Families: UltraScale, UltraScale+



### Introduction

In component mode, the OSERDESE3 in UltraScale devices is a dedicated parallel-to-serial converter with specific clocking and logic features designed to facilitate the implementation of high-speed source-synchronous applications. The OSERDESE3 avoids the additional timing complexities encountered when designing serializers in the internal device logic. The OSERDESE3 can serialize an outgoing signal by 4 in SDR mode, or by 4 and 8 in DDR mode.

### Port Descriptions

Port	Direction	Width	Function
CLK	Input	1	The high-speed clock input (CLK) is used to clock out the output serial data stream.
CLKDIV	Input	1	The divided clock input (CLKDIV) is typically a divided version of CLK (depending on the width of the implemented serialization). It drives the input of the parallel-to-serial converter and the CE module.
D<7:0>	Input	8	The parallel input data port (D) is the parallel data input port of the OSERDESE3.
OQ	Output	1	Serial Output Data to the IOB
RST	Input	1	Asynchronous Reset, active level based on IS_RST_INVERTED
T	Input	1	Tristate input from fabric
T_OUT	Output	1	3-state control output to IOB

### Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	Yes



## Available Attributes

Attribute	Type	Allowed Values	Default	Description
DATA_WIDTH	DECIMAL	8, 4	8	Defines the width of the parallel-to-serial converter. When used with SDR clocking, the DATA_WIDTH is set to be twice the desired width.
INIT	BINARY	1'b0, 1'b1	1'b0	Initializes the OSERDES flip-flops to the value specified
IS_CLK_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether the CLK pin is active-high or active-low.
IS_CLKDIV_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether the CLKDIV pin is active-high or active-low.
IS_RST_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether the RST pin is active-high or active-low.
ODDR_MODE	STRING	"FALSE", "TRUE"	"FALSE"	Internal property for Vivado primitive mapping. Do not modify.
SIM_DEVICE	STRING	"ULTRASCALE", "ULTRASCALE_PLUS_ES1"	"ULTRASCALE"	Set the device version

## VHDL Instantiation Template

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- OSERDESE3: Output SERIAL/DESerializer
--           UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

OSERDESE3_inst : OSERDESE3
generic map (
    DATA_WIDTH => 8,           -- Parallel Data Width (4-8)
    INIT => '0',               -- Initialization value of the OSERDES flip-flops
    IS_CLKDIV_INVERTED => '0', -- Optional inversion for CLKDIV
    IS_CLK_INVERTED => '0',    -- Optional inversion for CLK
    IS_RST_INVERTED => '0',    -- Optional inversion for RST
    SIM_DEVICE => "ULTRASCALE" -- Set the device version (ULTRASCALE, ULTRASCALE_PLUS_ES1)
)
port map (
    OQ => OQ,                 -- 1-bit output: Serial Output Data
    T_OUT => T_OUT,           -- 1-bit output: 3-state control output to IOB
    CLK => CLK,               -- 1-bit input: High-speed clock
    CLKDIV => CLKDIV,         -- 1-bit input: Divided Clock
    D => D,                   -- 8-bit input: Parallel Data Input
    RST => RST,               -- 1-bit input: Asynchronous Reset
    T => T                    -- 1-bit input: Tristate input from fabric
);

-- End of OSERDESE3_inst instantiation
    
```

## Verilog Instantiation Template

```
// OSERDESE3: Output SERIAL/DESerializer
//      UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

OSERDESE3 #(
    .DATA_WIDTH(8),           // Parallel Data Width (4-8)
    .INIT(1'b0),             // Initialization value of the OSERDES flip-flops
    .IS_CLKDIV_INVERTED(1'b0), // Optional inversion for CLKDIV
    .IS_CLK_INVERTED(1'b0),  // Optional inversion for CLK
    .IS_RST_INVERTED(1'b0),  // Optional inversion for RST
    .SIM_DEVICE("ULTRASCALE") // Set the device version (ULTRASCALE, ULTRASCALE_PLUS_ES1)
)
OSERDESE3_inst (
    .OQ(OQ),                 // 1-bit output: Serial Output Data
    .T_OUT(T_OUT),          // 1-bit output: 3-state control output to IOB
    .CLK(CLK),               // 1-bit input: High-speed clock
    .CLKDIV(CLKDIV),        // 1-bit input: Divided Clock
    .D(D),                   // 8-bit input: Parallel Data Input
    .RST(RST),               // 1-bit input: Asynchronous Reset
    .T(T)                    // 1-bit input: Tristate input from fabric
);

// End of OSERDESE3_inst instantiation
```

## For More Information

- See the *UltraScale Architecture SelectIO Resources User Guide* ([UG571](#)).
- See the [UltraScale Programmable Devices Documentation](#).

# PCIE40E4

Primitive: Integrated Block for PCI Express

PRIMITIVE\_GROUP: **ADVANCED**

PRIMITIVE\_SUBGROUP: PCIE

Families: UltraScale+

## Introduction

The Integrated block for PCI Express is a hard macro primitive compliant with the PCIe specification. This block is designed to be integrated with GTs and device clocking resources using fabric interconnect. See the Integrated Block for PCI Express User Guide for further details on this component.

## Design Entry Method

Instantiation	No
Inference	No
IP and IP Integrator Catalog	Recommended

## For More Information

- See the *UltraScale Architecture Gen3 Integrated Block for PCI Express v4.1 Product Guide (PG156)*.
- See the [UltraScale Programmable Devices Documentation](#).

# PCIE\_3\_1

Primitive: Integrated Block for PCI Express

PRIMITIVE\_GROUP: **ADVANCED**

PRIMITIVE\_SUBGROUP: PCIE

Families: UltraScale

## Introduction

The Integrated block for PCI Express is a hard macro primitive compliant with the PCIe specification. This block is designed to be integrated with GTs and device clocking resources using fabric interconnect. See the Integrated Block for PCI Express User Guide for further details on this component.

## Design Entry Method

Instantiation	No
Inference	No
IP and IP Integrator Catalog	Recommended

## For More Information

- See the *UltraScale Architecture Gen3 Integrated Block for PCI Express v4.1 Product Guide (PG156)*.
- See the [UltraScale Programmable Devices Documentation](#).

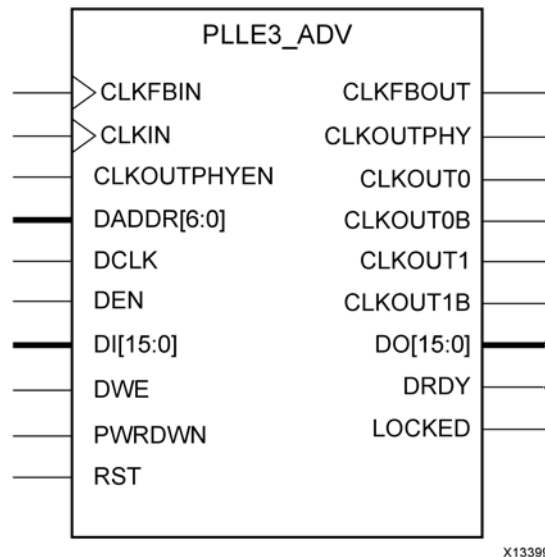
## PLLE3\_ADV

Primitive: Advanced Phase-Locked Loop (PLL)

PRIMITIVE\_GROUP: CLOCK

PRIMITIVE\_SUBGROUP: PLL

Families: UltraScale, UltraScale+



### Introduction

The PLLE3 is used for high-speed I/O clocking using Bitslice components as well as general clocking requirements. In general, the PLLE3 has less jitter and reduced power characteristics compared to the MMCME3 which makes it preferable for clocking behaviors that do not require features only available to the MMCME3.

### Port Descriptions

Port	Direction	Width	Function
CLKIN	Input	1	Clock input.
LOCKED	Output	1	An output from the PLL that indicates when the PLL has achieved phase alignment within a predefined window and frequency matching within a predefined PPM range. The PLL 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 PLL must be reset after LOCKED is deasserted.
<b>Clock Outputs:</b> User configurable clock outputs that can be divided versions of the VCO phase outputs (user controllable) from 1 (bypassed) to 256. The output clocks are phase aligned to each other (unless phase shifted) and aligned to the input clock with a proper feedback configuration.			
CLKOUTPHY	Output	1	General clock output connected to I/O Bitslice components.
CLKOUT0	Output	1	General clock output CLKOUT0. Generally connected to a global buffer.

Port	Direction	Width	Function
CLKOUT0B	Output	1	Inverted CLKOUT0. Generally connected to a global buffer.
CLKOUT1	Output	1	General clock output CLKOUT1. Generally connected to a global buffer.
CLKOUT1B	Output	1	Inverted CLKOUT1. Generally connected to a global buffer.
<b>Control Ports:</b> PLL control ports.			
CLKOUTPHYEN	Input	1	Enable signal for CLKOUTPHY.
PWRDWN	Input	1	Powers down instantiated but unused PLLs.
RST	Input	1	The RST signal is an asynchronous reset for the PLL. The PLL will synchronously re-enable itself when this signal is released and go through a new phase alignment and lock cycle. A reset is required when the input clock conditions change (e.g., frequency).
<b>DRP Ports:</b> Ports used when using the dynamic reconfigurable ports for reading and writing the configuration of the PLL.			
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 zeroes.
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 output bus provides PLL data output when using dynamic reconfiguration.
DO<15:0>	Output	16	The dynamic reconfiguration output bus provides PLL data output when using dynamic reconfiguration.
DRDY	Output	1	The dynamic reconfiguration ready output (DRDY) provides the response to the DEN signal for the PLLs 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.
<b>Feedback Clocks:</b> Required ports to form the feedback path for the PLL phase alignment capabilities.			
CLKFBIN	Input	1	Feedback clock pin to the PLL
CLKFBOUT	Output	1	Dedicated PLL Feedback clock output

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	Recommended

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
CLKFBOUT_MULT	DECIMAL	1 to 19	5	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 PLL.
CLKIN_PERIOD	FLOAT(ns)	0.000 to 14.286	0.000	Specifies the input period in ns to the PLL CLKIN input. Resolution is down to the ps (3 decimal places). For example a value of 33.333 would indicate a 30 MHz input clock. This information is mandatory and must be supplied.
CLKOUTPHY_MODE	STRING	"VCO_2X", "VCO", "VCO_HALF"	"VCO_2X"	Specifies the frequency of the CLKOUTPHY clock output.
COMPENSATION	STRING	"AUTO", "BUF_IN", "INTERNAL"	"AUTO"	<p>Clock input compensation. In general, should be set to AUTO. Defines how the PLL feedback compensation is configured.</p> <ul style="list-style-type: none"> <li>"AUTO" - Tools automatically determine proper compensation settings based on how the PLL feedback path is connected.</li> <li>"INTERNAL" - Indicates the PLL is using its own internal feedback path so no delay is being compensated.</li> <li>"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 for instance.</li> </ul>
DIVCLK_DIVIDE	DECIMAL	1 to 15	1	Specifies the division ratio for all output clocks with respect to the input clock. Effectively divides the CLKIN going into the PFD.
REF_JITTER	3 significant digit FLOAT	0.000 to 0.999	0.010	Allows specification of the expected jitter on CLKIN in order to better optimize PLL 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.

Attribute	Type	Allowed Values	Default	Description
STARTUP_WAIT	STRING	"FALSE", "TRUE"	"FALSE"	Delays configuration DONE signal from asserting until PLL is locked.
<b>CLKOUT0 Attributes:</b> Sets the Divide, Phase and Duty Cycle for the CLKOUT0 output				
CLKOUT0_DIVIDE	DECIMAL	1 to 128	1	Specifies the amount to divide the CLKOUT0 output. This number in combination with the CLKFBOUT_MULT and DIVCLK_DIVIDE values will determine the output frequency.
CLKOUT0_DUTY_CYCLE	3 significant digit FLOAT	0.001 to 0.999	0.500	Specifies the Duty Cycle of CLKOUT0 clock output in percentage (i.e., 0.500 will generate a 50% duty cycle).
CLKOUT0_PHASE	3 significant digit FLOAT	-360.000 to 360.000	0.000	Specifies the phase offset in degrees of the CLKOUT0 output.
<b>CLKOUT1 Attributes:</b> Sets the Divide, Phase and Duty Cycle for the CLKOUT1 output				
CLKOUT1_DIVIDE	DECIMAL	1 to 128	1	Specifies the amount to divide the CLKOUT1 output. This number in combination with the CLKFBOUT_MULT and DIVCLK_DIVIDE values will determine the output frequency.
CLKOUT1_DUTY_CYCLE	3 significant digit FLOAT	0.001 to 0.999	0.500	Specifies the Duty Cycle of CLKOUT1 clock output in percentage (i.e., 0.500 will generate a 50% duty cycle).
CLKOUT1_PHASE	3 significant digit FLOAT	-360.000 to 360.000	0.000	Specifies the phase offset in degrees of the CLKOUT1 output.
<b>Programmable Inversion Attributes:</b> Specifies whether or not to use the optional inversions on specific pins of this component to change the active polarity of the pin function. When set to 1 on a clock input pin (CLKIN or CLKFBIN), the phase is effectively shifted 180 degrees. When set to 1 on other pins, it changes the function to behave active-Low rather than active-High. If an external inverter is specified on one of these associated pins, the Vivado Design Suite will automatically set this attribute during the opt_design stage so that additional logic is not necessary for changing the input polarity.				
IS_CLKFBIN_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the CLKFBIN pin of this component.
IS_CLKIN_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the CLKIN pin of this component.
IS_PWRDWN_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the PWRDWN pin of this component.
IS_RST_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the RST pin of this component.



## VHDL Instantiation Template

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- PLLE3_ADV: Advanced Phase-Locked Loop (PLL)
-- UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

PLLE3_ADV_inst : PLLE3_ADV
generic map (
    CLKFBOUT_MULT => 5,          -- Multiply value for all CLKOUT, (1-19)
    CLKFBOUT_PHASE => 0.0,      -- Phase offset in degrees of CLKFB, (-360.000-360.000)
    CLKIN_PERIOD => 0.0,        -- Input clock period in ns to ps resolution (i.e. 33.333 is 30 MHz).
    -- CLKOUT0 Attributes: Divide, Phase and Duty Cycle for the CLKOUT0 output
    CLKOUT0_DIVIDE => 1,        -- Divide amount for CLKOUT0 (1-128)
    CLKOUT0_DUTY_CYCLE => 0.5,  -- Duty cycle for CLKOUT0 (0.001-0.999)
    CLKOUT0_PHASE => 0.0,      -- Phase offset for CLKOUT0 (-360.000-360.000)
    -- CLKOUT1 Attributes: Divide, Phase and Duty Cycle for the CLKOUT1 output
    CLKOUT1_DIVIDE => 1,        -- Divide amount for CLKOUT1 (1-128)
    CLKOUT1_DUTY_CYCLE => 0.5,  -- Duty cycle for CLKOUT1 (0.001-0.999)
    CLKOUT1_PHASE => 0.0,      -- Phase offset for CLKOUT1 (-360.000-360.000)
    CLKOUTPHY_MODE => "VCO_2X", -- Frequency of the CLKOUTPHY (VCO, VCO_2X, VCO_HALF)
    COMPENSATION => "AUTO",     -- AUTO, BUF_IN, INTERNAL
    DIVCLK_DIVIDE => 1,        -- Master division value, (1-15)
    -- Programmable Inversion Attributes: Specifies built-in programmable inversion on specific pins
    IS_CLKFBIN_INVERTED => '0', -- Optional inversion for CLKFBIN
    IS_CLKIN_INVERTED => '0',  -- Optional inversion for CLKIN
    IS_PWRDWN_INVERTED => '0', -- Optional inversion for PWRDWN
    IS_RST_INVERTED => '0',   -- Optional inversion for RST
    REF_JITTER => 0.0,        -- Reference input jitter in UI (0.000-0.999)
    STARTUP_WAIT => "FALSE"   -- Delays DONE until PLL is locked (FALSE, TRUE)
)
port map (
    -- Clock Outputs outputs: User configurable clock outputs
    CLKOUT0 => CLKOUT0,        -- 1-bit output: General Clock output
    CLKOUT0B => CLKOUT0B,     -- 1-bit output: Inverted CLKOUT0
    CLKOUT1 => CLKOUT1,       -- 1-bit output: General Clock output
    CLKOUT1B => CLKOUT1B,     -- 1-bit output: Inverted CLKOUT1
    CLKOUTPHY => CLKOUTPHY,   -- 1-bit output: Bitslice clock
    -- DRP Ports outputs: Dynamic reconfiguration ports
    DO => DO,                  -- 16-bit output: DRP data
    DRDY => DRDY,              -- 1-bit output: DRP ready
    -- Feedback Clocks outputs: Clock feedback ports
    CLKFBOUT => CLKFBOUT,     -- 1-bit output: Feedback clock
    LOCKED => LOCKED,         -- 1-bit output: LOCK
    CLKIN => CLKIN,           -- 1-bit input: Input clock
    -- Control Ports inputs: PLL control ports
    CLKOUTPHYEN => CLKOUTPHYEN, -- 1-bit input: CLKOUTPHY enable
    PWRDWN => PWRDWN,         -- 1-bit input: Power-down
    RST => RST,                -- 1-bit input: Reset
    -- DRP Ports inputs: Dynamic reconfiguration ports
    DADDR => DADDR,           -- 7-bit input: DRP address
    DCLK => DCLK,             -- 1-bit input: DRP clock
    DEN => DEN,               -- 1-bit input: DRP enable
    DI => DI,                  -- 16-bit input: DRP data
    DWE => DWE,               -- 1-bit input: DRP write enable
    -- Feedback Clocks inputs: Clock feedback ports
    CLKFBIN => CLKFBIN        -- 1-bit input: Feedback clock
);

-- End of PLLE3_ADV_inst instantiation
    
```

## Verilog Instantiation Template

```
// PLLE3_ADV: Advanced Phase-Locked Loop (PLL)
// UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

PLLE3_ADV #(
    .CLKFBOUT_MULT(5),           // Multiply value for all CLKOUT, (1-19)
    .CLKFBOUT_PHASE(0.0),       // Phase offset in degrees of CLKFB, (-360.000-360.000)
    .CLKIN_PERIOD(0.0),         // Input clock period in ns to ps resolution (i.e. 33.333 is 30 MHz).
    // CLKOUT0 Attributes: Divide, Phase and Duty Cycle for the CLKOUT0 output
    .CLKOUT0_DIVIDE(1),         // Divide amount for CLKOUT0 (1-128)
    .CLKOUT0_DUTY_CYCLE(0.5),   // Duty cycle for CLKOUT0 (0.001-0.999)
    .CLKOUT0_PHASE(0.0),        // Phase offset for CLKOUT0 (-360.000-360.000)
    // CLKOUT1 Attributes: Divide, Phase and Duty Cycle for the CLKOUT1 output
    .CLKOUT1_DIVIDE(1),         // Divide amount for CLKOUT1 (1-128)
    .CLKOUT1_DUTY_CYCLE(0.5),   // Duty cycle for CLKOUT1 (0.001-0.999)
    .CLKOUT1_PHASE(0.0),        // Phase offset for CLKOUT1 (-360.000-360.000)
    .CLKOUTPHY_MODE("VCO_2X"), // Frequency of the CLKOUTPHY (VCO, VCO_2X, VCO_HALF)
    .COMPENSATION("AUTO"),     // AUTO, BUF_IN, INTERNAL
    .DIVCLK_DIVIDE(1),          // Master division value, (1-15)
    // Programmable Inversion Attributes: Specifies built-in programmable inversion on specific pins
    .IS_CLKFBIN_INVERTED(1'b0), // Optional inversion for CLKFBIN
    .IS_CLKIN_INVERTED(1'b0),   // Optional inversion for CLKIN
    .IS_PWRDWN_INVERTED(1'b0), // Optional inversion for PWRDWN
    .IS_RST_INVERTED(1'b0),     // Optional inversion for RST
    .REF_JITTER(0.0),           // Reference input jitter in UI (0.000-0.999)
    .STARTUP_WAIT("FALSE")      // Delays DONE until PLL is locked (FALSE, TRUE)
)
PLLE3_ADV_inst (
    // Clock Outputs outputs: User configurable clock outputs
    .CLKOUT0(CLKOUT0),          // 1-bit output: General Clock output
    .CLKOUT0B(CLKOUT0B),       // 1-bit output: Inverted CLKOUT0
    .CLKOUT1(CLKOUT1),         // 1-bit output: General Clock output
    .CLKOUT1B(CLKOUT1B),       // 1-bit output: Inverted CLKOUT1
    .CLKOUTPHY(CLKOUTPHY),     // 1-bit output: Bitslice clock
    // DRP Ports outputs: Dynamic reconfiguration ports
    .DO(DO),                    // 16-bit output: DRP data
    .DRDY(DRDY),                // 1-bit output: DRP ready
    // Feedback Clocks outputs: Clock feedback ports
    .CLKFBOUT(CLKFBOUT),        // 1-bit output: Feedback clock
    .LOCKED(LOCKED),            // 1-bit output: LOCK
    .CLKIN(CLKIN),              // 1-bit input: Input clock
    // Control Ports inputs: PLL control ports
    .CLKOUTPHYEN(CLKOUTPHYEN), // 1-bit input: CLKOUTPHY enable
    .PWRDWN(PWRDWN),           // 1-bit input: Power-down
    .RST(RST),                  // 1-bit input: Reset
    // DRP Ports inputs: Dynamic reconfiguration ports
    .DADDR(DADDR),              // 7-bit input: DRP address
    .DCLK(DCLK),                // 1-bit input: DRP clock
    .DEN(DEN),                  // 1-bit input: DRP enable
    .DI(DI),                    // 16-bit input: DRP data
    .DWE(DWE),                  // 1-bit input: DRP write enable
    // Feedback Clocks inputs: Clock feedback ports
    .CLKFBIN(CLKFBIN)           // 1-bit input: Feedback clock
);

// End of PLLE3_ADV_inst instantiation
```

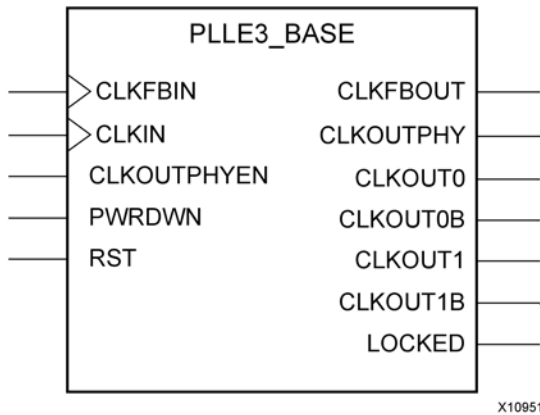
## For More Information

- See the *UltraScale Architecture Clocking Resources User Guide* ([UG572](#)).
- See the [UltraScale Programmable Devices Documentation](#).

# PLLE3\_BASE

Primitive: Base Phase-Locked Loop (PLL)

PRIMITIVE\_GROUP: CLOCK  
 PRIMITIVE\_SUBGROUP: PLL  
 Families: UltraScale, UltraScale+



## Introduction

The PLLE3 is used for high-speed I/O clocking using Bitslice components as well as general clocking requirements. In general, the PLLE3 has less jitter and reduced power characteristics compared to the MMCME3 which makes it preferable for clocking behaviors that do not require features only available to the MMCME3.

## Port Descriptions

Port	Direction	Width	Function
CLKIN	Input	1	Clock input.
LOCKED	Output	1	An output from the PLL that indicates when the PLL has achieved phase alignment within a predefined window and frequency matching within a predefined PPM range. The PLL 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 PLL must be reset after LOCKED is deasserted.
<b>Clock Outputs:</b> User configurable clock outputs that can be divided versions of the VCO phase outputs (user controllable) from 1 (bypassed) to 256. The output clocks are phase aligned to each other (unless phase shifted) and aligned to the input clock with a proper feedback configuration.			
CLKOUTPHY	Output	1	General clock output connected to I/O Bitslice components.
CLKOUT0	Output	1	General clock output CLKOUT0. Generally connected to a global buffer.
CLKOUT0B	Output	1	Inverted CLKOUT0. Generally connected to a global buffer.
CLKOUT1	Output	1	General clock output CLKOUT1. Generally connected to a global buffer.

Port	Direction	Width	Function
CLKOUT1B	Output	1	Inverted CLKOUT1. Generally connected to a global buffer.
<b>Control Ports:</b> PLL control ports.			
CLKOUTPHYEN	Input	1	Enable signal for CLKOUTPHY.
PWRDWN	Input	1	Powers down instantiated but unused PLLs.
RST	Input	1	The RST signal is an asynchronous reset for the PLL. The PLL will synchronously re-enable itself when this signal is released and go through a new phase alignment and lock cycle. A reset is required when the input clock conditions change (e.g., frequency).
<b>Feedback Clocks:</b> Required ports to form the feedback path for the PLL phase alignment capabilities.			
CLKFBIN	Input	1	Feedback clock pin to the PLL
CLKFBOUT	Output	1	Dedicated PLL Feedback clock output

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	Recommended

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
CLKFBOUT_MULT	DECIMAL	1 to 19	5	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 PLL.
CLKIN_PERIOD	FLOAT(ns)	0.000 to 14.286	0.000	Specifies the input period in ns to the PLL CLKIN input. Resolution is down to the ps (3 decimal places). For example a value of 33.333 would indicate a 30 MHz input clock. This information is mandatory and must be supplied.
CLKOUTPHY_MODE	STRING	"VCO_2X", "VCO", "VCO_HALF"	"VCO_2X"	Specifies the frequency of the CLKOUTPHY clock output.
DIVCLK_DIVIDE	DECIMAL	1 to 15	1	Specifies the division ratio for all output clocks with respect to the input clock. Effectively divides the CLKIN going into the PFD.

Attribute	Type	Allowed Values	Default	Description
REF_JITTER	3 significant digit FLOAT	0.000 to 0.999	0.010	Allows specification of the expected jitter on CLKIN in order to better optimize PLL 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	STRING	"FALSE", "TRUE"	"FALSE"	Delays configuration DONE signal from asserting until PLL is locked.
<b>CLKOUT0 Attributes:</b> Sets the Divide, Phase and Duty Cycle for the CLKOUT0 output				
CLKOUT0_DIVIDE	DECIMAL	1 to 128	1	Specifies the amount to divide the CLKOUT0 output. This number in combination with the CLKFBOUT_MULT and DIVCLK_DIVIDE values will determine the output frequency.
CLKOUT0_DUTY_CYCLE	3 significant digit FLOAT	0.001 to 0.999	0.500	Specifies the Duty Cycle of CLKOUT0 clock output in percentage (i.e., 0.500 will generate a 50% duty cycle).
CLKOUT0_PHASE	3 significant digit FLOAT	-360.000 to 360.000	0.000	Specifies the phase offset in degrees of the CLKOUT0 output.
<b>CLKOUT1 Attributes:</b> Sets the Divide, Phase and Duty Cycle for the CLKOUT1 output				
CLKOUT1_DIVIDE	DECIMAL	1 to 128	1	Specifies the amount to divide the CLKOUT1 output. This number in combination with the CLKFBOUT_MULT and DIVCLK_DIVIDE values will determine the output frequency.
CLKOUT1_DUTY_CYCLE	3 significant digit FLOAT	0.001 to 0.999	0.500	Specifies the Duty Cycle of CLKOUT1 clock output in percentage (i.e., 0.500 will generate a 50% duty cycle).
CLKOUT1_PHASE	3 significant digit FLOAT	-360.000 to 360.000	0.000	Specifies the phase offset in degrees of the CLKOUT1 output.
<b>Programmable Inversion Attributes:</b> Specifies whether or not to use the optional inversions on specific pins of this component to change the active polarity of the pin function. When set to 1 on a clock input pin (CLKIN or CLKFBIN), the phase is effectively shifted 180 degrees. When set to 1 on other pins, it changes the function to behave active-Low rather than active-High. If an external inverter is specified on one of these associated pins, the Vivado Design Suite will automatically set this attribute during the opt_design stage so that additional logic is not necessary for changing the input polarity.				
IS_CLKFBIN_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the CLKFBIN pin of this component.
IS_CLKIN_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the CLKIN pin of this component.

Attribute	Type	Allowed Values	Default	Description
IS_PWRDWN_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the PWRDWN pin of this component.
IS_RST_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the RST pin of this component.

## VHDL Instantiation Template

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- PLLE3_BASE: Base Phase-Locked Loop (PLL)
-- UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

PLLE3_BASE_inst : PLLE3_BASE
generic map (
  CLKFBOUT_MULT => 5,          -- Multiply value for all CLKOUT, (1-19)
  CLKFBOUT_PHASE => 0.0,      -- Phase offset in degrees of CLKFB, (-360.000-360.000)
  CLKIN_PERIOD => 0.0,        -- Input clock period in ns to ps resolution (i.e. 33.333 is 30 MHz).
  -- CLKOUT0 Attributes: Divide, Phase and Duty Cycle for the CLKOUT0 output
  CLKOUT0_DIVIDE => 1,        -- Divide amount for CLKOUT0 (1-128)
  CLKOUT0_DUTY_CYCLE => 0.5,  -- Duty cycle for CLKOUT0 (0.001-0.999)
  CLKOUT0_PHASE => 0.0,      -- Phase offset for CLKOUT0 (-360.000-360.000)
  -- CLKOUT1 Attributes: Divide, Phase and Duty Cycle for the CLKOUT1 output
  CLKOUT1_DIVIDE => 1,        -- Divide amount for CLKOUT1 (1-128)
  CLKOUT1_DUTY_CYCLE => 0.5,  -- Duty cycle for CLKOUT1 (0.001-0.999)
  CLKOUT1_PHASE => 0.0,      -- Phase offset for CLKOUT1 (-360.000-360.000)
  CLKOUTPHY_MODE => "VCO_2X", -- Frequency of the CLKOUTPHY (VCO, VCO_2X, VCO_HALF)
  DIVCLK_DIVIDE => 1,        -- Master division value, (1-15)
  -- Programmable Inversion Attributes: Specifies built-in programmable inversion on specific pins
  IS_CLKFBIN_INVERTED => '0', -- Optional inversion for CLKFBIN
  IS_CLKIN_INVERTED => '0',  -- Optional inversion for CLKIN
  IS_PWRDWN_INVERTED => '0', -- Optional inversion for PWRDWN
  IS_RST_INVERTED => '0',   -- Optional inversion for RST
  REF_JITTER => 0.0,        -- Reference input jitter in UI (0.000-0.999)
  STARTUP_WAIT => "FALSE"   -- Delays DONE until PLL is locked (FALSE, TRUE)
)
port map (
  -- Clock Outputs outputs: User configurable clock outputs
  CLKOUT0 => CLKOUT0,       -- 1-bit output: General Clock output
  CLKOUT0B => CLKOUT0B,    -- 1-bit output: Inverted CLKOUT0
  CLKOUT1 => CLKOUT1,      -- 1-bit output: General Clock output
  CLKOUT1B => CLKOUT1B,    -- 1-bit output: Inverted CLKOUT1
  CLKOUTPHY => CLKOUTPHY,  -- 1-bit output: Bitslice clock
  -- Feedback Clocks outputs: Clock feedback ports
  CLKFBOUT => CLKFBOUT,    -- 1-bit output: Feedback clock
  LOCKED => LOCKED,        -- 1-bit output: LOCK
  CLKIN => CLKIN,          -- 1-bit input: Input clock
  -- Control Ports inputs: PLL control ports
  CLKOUTPHYEN => CLKOUTPHYEN, -- 1-bit input: CLKOUTPHY enable
  PWRDWN => PWRDWN,        -- 1-bit input: Power-down
  RST => RST,              -- 1-bit input: Reset
  -- Feedback Clocks inputs: Clock feedback ports
  CLKFBIN => CLKFBIN      -- 1-bit input: Feedback clock
);

-- End of PLLE3_BASE_inst instantiation

```

## Verilog Instantiation Template

```
// PLLE3_BASE: Base Phase-Locked Loop (PLL)
// UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

PLLE3_BASE #(
    .CLKFBOUT_MULT(5),           // Multiply value for all CLKOUT, (1-19)
    .CLKFBOUT_PHASE(0.0),       // Phase offset in degrees of CLKFB, (-360.000-360.000)
    .CLKIN_PERIOD(0.0),         // Input clock period in ns to ps resolution (i.e. 33.333 is 30 MHz).
    // CLKOUT0 Attributes: Divide, Phase and Duty Cycle for the CLKOUT0 output
    .CLKOUT0_DIVIDE(1),         // Divide amount for CLKOUT0 (1-128)
    .CLKOUT0_DUTY_CYCLE(0.5),    // Duty cycle for CLKOUT0 (0.001-0.999)
    .CLKOUT0_PHASE(0.0),        // Phase offset for CLKOUT0 (-360.000-360.000)
    // CLKOUT1 Attributes: Divide, Phase and Duty Cycle for the CLKOUT1 output
    .CLKOUT1_DIVIDE(1),         // Divide amount for CLKOUT1 (1-128)
    .CLKOUT1_DUTY_CYCLE(0.5),    // Duty cycle for CLKOUT1 (0.001-0.999)
    .CLKOUT1_PHASE(0.0),        // Phase offset for CLKOUT1 (-360.000-360.000)
    .CLKOUTPHY_MODE("VCO_2X"),  // Frequency of the CLKOUTPHY (VCO, VCO_2X, VCO_HALF)
    .DIVCLK_DIVIDE(1),          // Master division value, (1-15)
    // Programmable Inversion Attributes: Specifies built-in programmable inversion on specific pins
    .IS_CLKFBIN_INVERTED(1'b0), // Optional inversion for CLKFBIN
    .IS_CLKIN_INVERTED(1'b0),   // Optional inversion for CLKIN
    .IS_PWRDWN_INVERTED(1'b0),  // Optional inversion for PWRDWN
    .IS_RST_INVERTED(1'b0),     // Optional inversion for RST
    .REF_JITTER(0.0),           // Reference input jitter in UI (0.000-0.999)
    .STARTUP_WAIT("FALSE")     // Delays DONE until PLL is locked (FALSE, TRUE)
)
PLLE3_BASE_inst (
    // Clock Outputs outputs: User configurable clock outputs
    .CLKOUT0(CLKOUT0),          // 1-bit output: General Clock output
    .CLKOUT0B(CLKOUT0B),        // 1-bit output: Inverted CLKOUT0
    .CLKOUT1(CLKOUT1),          // 1-bit output: General Clock output
    .CLKOUT1B(CLKOUT1B),        // 1-bit output: Inverted CLKOUT1
    .CLKOUTPHY(CLKOUTPHY),      // 1-bit output: Bitslice clock
    // Feedback Clocks outputs: Clock feedback ports
    .CLKFBOUT(CLKFBOUT),        // 1-bit output: Feedback clock
    .LOCKED(LOCKED),           // 1-bit output: LOCK
    .CLKIN(CLKIN),              // 1-bit input: Input clock
    // Control Ports inputs: PLL control ports
    .CLKOUTPHYEN(CLKOUTPHYEN), // 1-bit input: CLKOUTPHY enable
    .PWRDWN(PWRDWN),           // 1-bit input: Power-down
    .RST(RST),                  // 1-bit input: Reset
    // Feedback Clocks inputs: Clock feedback ports
    .CLKFBIN(CLKFBIN)          // 1-bit input: Feedback clock
);

// End of PLLE3_BASE_inst instantiation
```

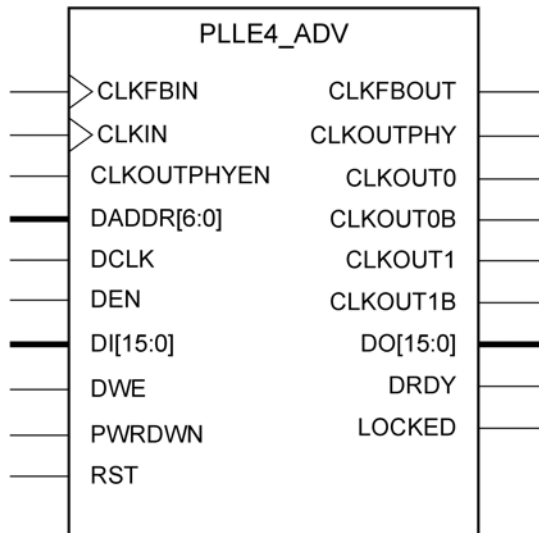
## For More Information

- See the *UltraScale Architecture Clocking Resources User Guide* ([UG572](#)).
- See the [UltraScale Programmable Devices Documentation](#).

## PLLE4\_ADV

Primitive: Advanced Phase-Locked Loop (PLL)

PRIMITIVE\_GROUP: CLOCK  
PRIMITIVE\_SUBGROUP: PLL  
Families: UltraScale+



X15110-102815

### Introduction

The PLLE4 is used for high-speed I/O clocking using Bitslice components as well as general clocking requirements. In general, the PLLE4 has less jitter and reduced power characteristics compared to the MMCME4 which makes it preferable for clocking behaviors that do not require features only available to the MMCME4.

### Port Descriptions

Port	Direction	Width	Function
CLKFBIN	Input	1	Feedback clock pin to the PLL
CLKFBOUT	Output	1	Dedicated PLL Feedback clock output
CLKIN	Input	1	Input clock
CLKOUTPHY	Output	1	General clock output connected to I/O Bitslice components
CLKOUTPHYEN	Input	1	Enable signal for CLKOUTPHY
CLKOUT0	Output	1	General clock output CLKOUT0. Generally connected to a global buffer
CLKOUT0B	Output	1	Inverted CLKOUT0
CLKOUT1	Output	1	General clock output CLKOUT1



Port	Direction	Width	Function
CLKOUT1B	Output	1	Inverted CLKOUT1
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 zeroes
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 output bus provides PLL data output when using dynamic reconfiguration
DO<15:0>	Output	16	The dynamic reconfiguration output bus provides PLL data output when using dynamic reconfiguration
DRDY	Output	1	The dynamic reconfiguration ready output (DRDY) provides the response to the DEN signal for the PLLs 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 PLL that indicates when the PLL has achieved phase alignment within a predefined window and frequency matching within a predefined PPM range. The PLL 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 PLL must be reset after LOCKED is deasserted
PWRDWN	Input	1	Powers down instantiated but unused PLLs
RST	Input	1	The RST signal is an asynchronous reset for the PLL. The PLL will synchronously re-enable itself when this signal is released and go through a new phase alignment and lock cycle. A reset is required when the input clock conditions change (e.g., frequency)

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	Recommended

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
CLKFBOUT_MULT	DECIMAL	2 to 21	5	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 PLL
CLKIN_PERIOD	FLOAT(ns)	0.000 to 14.286	0.000	Specifies the input period in ns to the PLL CLKIN input. Resolution is down to the ps (3 decimal places). For example a value of 33.333 would indicate a 30 MHz input clock. This information is mandatory and must be supplied
CLKOUTPHY_MODE	STRING	"VCO_2X", "VCO", "VCO_HALF"	"VCO_2X"	Specifies the frequency of the CLKOUTPHY clock output
CLKOUT0_DIVIDE	DECIMAL	1 to 128	1	Specifies the amount to divide the CLKOUT0 output. This number in combination with the CLKFBOUT_MULT and DIVCLK_DIVIDE values will determine the output frequency
CLKOUT0_DUTY_CYCLE	3 significant digit FLOAT	0.001 to 0.999	0.500	Specifies the Duty Cycle of CLKOUT0 clock output in percentage (i.e., 0.500 will generate a 50% duty cycle)
CLKOUT0_PHASE	3 significant digit FLOAT	-360.000 to 360.000	0.000	Specifies the phase offset in degrees of the CLKOUT0 output
CLKOUT1_DIVIDE	DECIMAL	1 to 128	1	Specifies the amount to divide the CLKOUT1 output. This number in combination with the CLKFBOUT_MULT and DIVCLK_DIVIDE values will determine the output frequency
CLKOUT1_DUTY_CYCLE	3 significant digit FLOAT	0.001 to 0.999	0.500	Specifies the Duty Cycle of CLKOUT1 clock output in percentage (i.e., 0.500 will generate a 50% duty cycle)
CLKOUT1_PHASE	3 significant digit FLOAT	-360.000 to 360.000	0.000	Specifies the phase offset in degrees of the CLKOUT1 output

Attribute	Type	Allowed Values	Default	Description
COMPENSATION	STRING	"AUTO", "BUF_IN", "INTERNAL", "PHY_ALIGN"	"AUTO"	<p>Clock input compensation. In general, should be set to AUTO. Defines how the PLL feedback compensation is configured.</p> <ul style="list-style-type: none"> <li>"AUTO" - Tools automatically determine proper compensation settings based on how the PLL feedback path is connected.</li> <li>"INTERNAL" - Indicates the PLL is using its own internal feedback path so no delay is being compensated.</li> <li>"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 for instance.</li> </ul>
DIVCLK_DIVIDE	DECIMAL	1 to 15	1	Specifies the division ratio for all output clocks with respect to the input clock. Effectively divides the CLKIN going into the PFD
IS_CLKFBIN_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the CLKFBIN pin of this component
IS_CLKIN_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the CLKIN pin of this component
IS_PWRDWN_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the PWRDWN pin of this component
IS_RST_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the RST pin of this component
REF_JITTER	3 significant digit FLOAT	0.000 to 0.999	0.010	Allows specification of the expected jitter on CLKIN in order to better optimize PLL 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	STRING	"FALSE", "TRUE"	"FALSE"	Delays configuration DONE signal from asserting until PLL is locked

## VHDL Instantiation Template

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- PLLE4_ADV: Advanced Phase-Locked Loop (PLL)
--           UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

PLLE4_ADV_inst : PLLE4_ADV
generic map (
    CLKFBOUT_MULT => 5,           -- Multiply value for all CLKOUT
    CLKFBOUT_PHASE => 0.0,       -- Phase offset in degrees of CLKFB
    CLKIN_PERIOD => 0.0,        -- Input clock period in ns to ps resolution (i.e. 33.333 is 30 MHz).
    CLKOUT0_DIVIDE => 1,        -- Divide amount for CLKOUT0
    CLKOUT0_DUTY_CYCLE => 0.5,   -- Duty cycle for CLKOUT0
    CLKOUT0_PHASE => 0.0,       -- Phase offset for CLKOUT0
    CLKOUT1_DIVIDE => 1,        -- Divide amount for CLKOUT1
    CLKOUT1_DUTY_CYCLE => 0.5,   -- Duty cycle for CLKOUT1
    CLKOUT1_PHASE => 0.0,       -- Phase offset for CLKOUT1
    CLKOUTPHY_MODE => "VCO_2X", -- Frequency of the CLKOUTPHY
    COMPENSATION => "AUTO",     -- Clock input compensation
    DIVCLK_DIVIDE => 1,        -- Master division value
    IS_CLKFBIN_INVERTED => '0', -- Optional inversion for CLKFBIN
    IS_CLKIN_INVERTED => '0',  -- Optional inversion for CLKIN
    IS_PWRDWN_INVERTED => '0', -- Optional inversion for PWRDWN
    IS_RST_INVERTED => '0',    -- Optional inversion for RST
    REF_JITTER => 0.0,        -- Reference input jitter in UI
    STARTUP_WAIT => "FALSE"    -- Delays DONE until PLL is locked
)
port map (
    CLKFBOUT => CLKFBOUT,       -- 1-bit output: Feedback clock
    CLKOUT0 => CLKOUT0,        -- 1-bit output: General Clock output
    CLKOUT0B => CLKOUT0B,      -- 1-bit output: Inverted CLKOUT0
    CLKOUT1 => CLKOUT1,       -- 1-bit output: General Clock output
    CLKOUT1B => CLKOUT1B,     -- 1-bit output: Inverted CLKOUT1
    CLKOUTPHY => CLKOUTPHY,    -- 1-bit output: Bitslice clock
    DO => DO,                  -- 16-bit output: DRP data output
    DRDY => DRDY,             -- 1-bit output: DRP ready
    LOCKED => LOCKED,         -- 1-bit output: LOCK
    CLKFBIN => CLKFBIN,       -- 1-bit input: Feedback clock
    CLKIN => CLKIN,           -- 1-bit input: Input clock
    CLKOUTPHYEN => CLKOUTPHYEN, -- 1-bit input: CLKOUTPHY enable
    DADDR => DADDR,           -- 7-bit input: DRP address
    DCLK => DCLK,             -- 1-bit input: DRP clock
    DEN => DEN,               -- 1-bit input: DRP enable
    DI => DI,                 -- 16-bit input: DRP data input
    DWE => DWE,               -- 1-bit input: DRP write enable
    PWRDWN => PWRDWN,        -- 1-bit input: Power-down
    RST => RST                 -- 1-bit input: Reset
);

-- End of PLLE4_ADV_inst instantiation
    
```

## Verilog Instantiation Template

```
// PLLE4_ADV: Advanced Phase-Locked Loop (PLL)
//      UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

PLLE4_ADV #(
    .CLKFBOUT_MULT(5),           // Multiply value for all CLKOUT
    .CLKFBOUT_PHASE(0.0),       // Phase offset in degrees of CLKFB
    .CLKIN_PERIOD(0.0),         // Input clock period in ns to ps resolution (i.e. 33.333 is 30 MHz).
    .CLKOUT0_DIVIDE(1),          // Divide amount for CLKOUT0
    .CLKOUT0_DUTY_CYCLE(0.5),    // Duty cycle for CLKOUT0
    .CLKOUT0_PHASE(0.0),        // Phase offset for CLKOUT0
    .CLKOUT1_DIVIDE(1),          // Divide amount for CLKOUT1
    .CLKOUT1_DUTY_CYCLE(0.5),    // Duty cycle for CLKOUT1
    .CLKOUT1_PHASE(0.0),        // Phase offset for CLKOUT1
    .CLKOUTPHY_MODE("VCO_2X"),   // Frequency of the CLKOUTPHY
    .COMPENSATION("AUTO"),       // Clock input compensation
    .DIVCLK_DIVIDE(1),           // Master division value
    .IS_CLKFBIN_INVERTED(1'b0),  // Optional inversion for CLKFBIN
    .IS_CLKIN_INVERTED(1'b0),    // Optional inversion for CLKIN
    .IS_PWRDWN_INVERTED(1'b0),  // Optional inversion for PWRDWN
    .IS_RST_INVERTED(1'b0),     // Optional inversion for RST
    .REF_JITTER(0.0),           // Reference input jitter in UI
    .STARTUP_WAIT("FALSE")      // Delays DONE until PLL is locked
)
PLLE4_ADV_inst (
    .CLKFBOUT(CLKFBOUT),         // 1-bit output: Feedback clock
    .CLKOUT0(CLKOUT0),           // 1-bit output: General Clock output
    .CLKOUT0B(CLKOUT0B),         // 1-bit output: Inverted CLKOUT0
    .CLKOUT1(CLKOUT1),           // 1-bit output: General Clock output
    .CLKOUT1B(CLKOUT1B),         // 1-bit output: Inverted CLKOUT1
    .CLKOUTPHY(CLKOUTPHY),       // 1-bit output: Bitslice clock
    .DO(DO),                     // 16-bit output: DRP data output
    .DRDY(DRDY),                 // 1-bit output: DRP ready
    .LOCKED(LOCKED),             // 1-bit output: LOCK
    .CLKFBIN(CLKFBIN),           // 1-bit input: Feedback clock
    .CLKIN(CLKIN),               // 1-bit input: Input clock
    .CLKOUTPHYEN(CLKOUTPHYEN),   // 1-bit input: CLKOUTPHY enable
    .DADDR(DADDR),               // 7-bit input: DRP address
    .DCLK(DCLK),                 // 1-bit input: DRP clock
    .DEN(DEN),                   // 1-bit input: DRP enable
    .DI(DI),                     // 16-bit input: DRP data input
    .DWE(DWE),                   // 1-bit input: DRP write enable
    .PWRDWN(PWRDWN),             // 1-bit input: Power-down
    .RST(RST)                    // 1-bit input: Reset
);

// End of PLLE4_ADV_inst instantiation
```

## For More Information

- See the *UltraScale Architecture Clocking Resources User Guide* ([UG572](#)).
- See the [UltraScale Programmable Devices Documentation](#).

# PULLDOWN

Primitive: I/O Pulldown

PRIMITIVE\_GROUP: I/O

PRIMITIVE\_SUBGROUP: WEAK\_DRIVER

Families: UltraScale, UltraScale+

PULLDOWN



X10690

## Introduction

The design element is a weak pulldown element that pulls an undriven I/O to a logic zero state. For example, if the I/O is 3-stated and not driven by any other element, a logic 0 will exist on the I/O.

## Port Descriptions

Port	Direction	Width	Function
O	Output	1	Pulldown output. Connect directly to a top-level port in the design.

## Design Entry Method

Instantiation	Yes
Inference	Yes, via property
IP and IP Integrator Catalog	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 Pulldown
-- UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

PULLDOWN_inst : PULLDOWN
port map (

```

```
    O => O -- 1-bit output: Pulldown output (connect directly to top-level port)
);
-- End of PULLDOWN_inst instantiation
```

## Verilog Instantiation Template

```
// PULLDOWN: I/O Pulldown
//           UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

PULLDOWN PULLDOWN_inst (
    .O(O) // 1-bit output: Pulldown output (connect directly to top-level port)
);

// End of PULLDOWN_inst instantiation
```

## For More Information

- See the *UltraScale Architecture SelectIO Resources User Guide* ([UG571](#)).
- See the [UltraScale Programmable Devices Documentation](#).

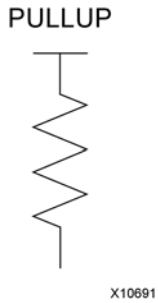
# PULLUP

Primitive: I/O Pullup

PRIMITIVE\_GROUP: I/O

PRIMITIVE\_SUBGROUP: WEAK\_DRIVER

Families: UltraScale, UltraScale+



## Introduction

The design element is a weak pullup element that pulls an undriven I/O to a logic one state. For example, if the I/O is 3-stated and not driven by any other element, a logic 1 will exist on the I/O.

## Port Descriptions

Port	Direction	Width	Function
O	Output	1	Pullup output. Connect directly to a top-level port in the design.

## Design Entry Method

Instantiation	Yes
Inference	Yes, via property
IP and IP Integrator Catalog	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 Pullup
-- UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

PULLUP_inst : PULLUP
port map (
    O => O -- 1-bit output: Pullup output (connect directly to top-level port)
);

```



```
-- End of PULLUP_inst instantiation
```

## Verilog Instantiation Template

```
// PULLUP: I/O Pullup
//      UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

PULLUP PULLUP_inst (
    .O(O) // 1-bit output: Pullup output (connect directly to top-level port)
);

// End of PULLUP_inst instantiation
```

## For More Information

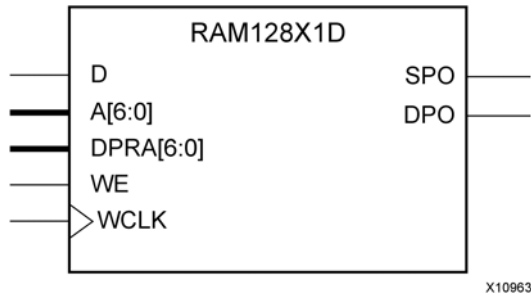
- See the *UltraScale Architecture SelectIO Resources User Guide* ([UG571](#)).
- See the [UltraScale Programmable Devices Documentation](#).

## RAM128X1D

Primitive: 128-Deep by 1-Wide Dual Port Random Access Memory (Select RAM)

PRIMITIVE\_GROUP: CLB

PRIMITIVE\_SUBGROUP: LUTRAM



### 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 memory cell 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 cell 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 outputting 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.
- Connect the WE clock enable pin to the proper write enable source in the design.
- Connect the 7-bit A bus to the source for the read/write addressing and the 7-bit DPRA bus to the appropriate read address connections.

You can use the INIT attribute to specify the initial contents of the RAM. If left unspecified, the initial contents default to all zeros.

## Design Entry Method

Instantiation	Yes
Inference	Recommended
IP Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	HEX	Any 128-bit value	All zeros	Specifies the initial contents of the RAM.
IS_WCLK_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the WCLK pin.

## VHDL 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
--           UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

RAM128X1D_inst : RAM128X1D
generic map (
  INIT => X"00000000000000000000000000000000", -- Initial contents of RAM
  IS_WCLK_INVERTED => '0') -- Specifies active high/low WCLK
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
//          UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

RAM128X1D #(
    .INIT(128'h00000000000000000000000000000000),
    .IS_WCLK_INVERTED(1'b0) // Specifies active high/low WCLK
) RAM128X1D_inst (
    .DPO(DPO), // Read port 1-bit output
    .SPO(SPO), // Read/write 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) // Write enable input
);

// End of RAM128X1D_inst instantiation
```

## For More Information

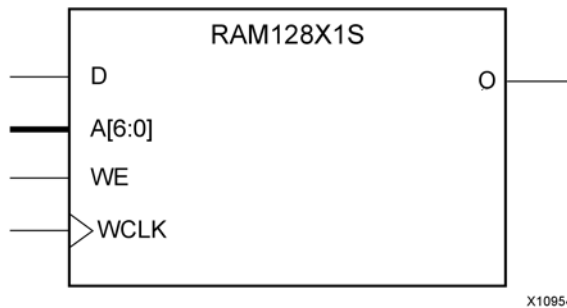
- See the *UltraScale Architecture Migration: Methodology Guide* ([UG1026](#)).
- See the [UltraScale Programmable Devices Documentation](#).

## RAM128X1S

Primitive: 128-Deep by 1-Wide Random Access Memory (Select RAM)

PRIMITIVE\_GROUP: CLB

PRIMITIVE\_SUBGROUP: LUTRAM



### Introduction

This design element is a 128-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 CLB.

The RAM128X1S 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 cell 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	7	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
IP Catalog	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.
- Connect the WE clock enable pin to the proper write enable source in the design.
- Connect the 7-bit A bus to the source for the read/write.

You can use the INIT attribute to specify the initial contents of the RAM. If left unspecified, the initial contents default to all zeros.

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	HEX	Any 128-bit value	All zeros	Specifies the initial contents of the RAM.
IS_WCLK_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the WCLK pin.

## VHDL Instantiation Template

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAM128X1S: 128-deep x 1 positive edge write, asynchronous read
--           single-port distributed RAM
--           UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

RAM128X1S_inst : RAM128X1S
generic map (
  INIT => X"00000000000000000000000000000000", -- Initial contents of RAM
  IS_WCLK_INVERTED => '0') -- Specifies active high/low WCLK
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
  A6 => A6,    -- Address[6] input bit
  D => D,      -- 1-bit data input
  WCLK => WCLK, -- Write clock input
  WE => WE     -- RAM data input
);

-- End of RAM128X1S_inst instantiation

```

## Verilog Instantiation Template

```
// RAM128X1S: 128 x 1 positive edge write, asynchronous read single-port
//           distributed RAM (Mapped to two LUT6s)
//           UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

RAM128X1S #(
    .INIT(128'h00000000000000000000000000000000), // Initial contents of RAM
    .IS_WCLK_INVERTED(1'b0) // Specifies active high/low WCLK
) RAM128X1S_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
    .A6(A6), // Address[6] input bit
    .D(D), // 1-bit data input
    .WCLK(WCLK), // Write clock input
    .WE(WE) // Write enable input
);

// End of RAM128X1S_inst instantiation
```

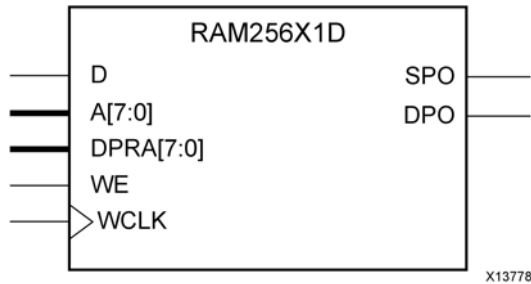
## For More Information

- See the *UltraScale Architecture Configurable Logic Block User Guide* ([UG574](#)).
- See the [UltraScale Programmable Devices Documentation](#).

# RAM256X1D

Primitive: 256-Deep by 1-Wide Dual Port Random Access Memory (Select RAM)

PRIMITIVE\_GROUP: CLB  
 PRIMITIVE\_SUBGROUP: LUTRAM



## Introduction

This design element is a 256-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 memory cell 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 cell 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 outputting 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	8	Read/Write port address bus
DPRA	Input	8	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.
- Connect the WE clock enable pin to the proper write enable source in the design.
- Connect the 7-bit A bus to the source for the read/write addressing and the 7-bit DPRA bus to the appropriate read address connections.



You can use the INIT attribute to specify the initial contents of the RAM. If left unspecified, the initial contents default to all zeros.

## Design Entry Method

Instantiation	Yes
Inference	Recommended
IP Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	HEX	Any 256-bit value	All zeros	Specifies the initial contents of the RAM.
IS_WCLK_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the WCLK pin.

## VHDL Instantiation Template

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAM256X1D: 256-deep by 1-wide positive edge write, asynchronous read
--           dual-port distributed LUT RAM
--           UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

RAM256X1D_inst : RAM256X1D
generic map (
  INIT => X"0000000000000000000000000000000000000000000000000000000000000000", -- Initial contents of RAM
  IS_WCLK_INVERTED => '0') -- Specifies active high/low WCLK
port map (
  DPO => DPO,      -- Read/Write port 1-bit output
  SPO => SPO,      -- Read port 1-bit output
  A => A,          -- Read/Write port 8-bit address input
  D => D,          -- RAM data input
  DPRA => DPRA,    -- Read port 8-bit address input
  WCLK => WCLK,    -- Write clock input
  WE => WE         -- RAM data input
);

-- End of RAM256X1D_inst instantiation

```

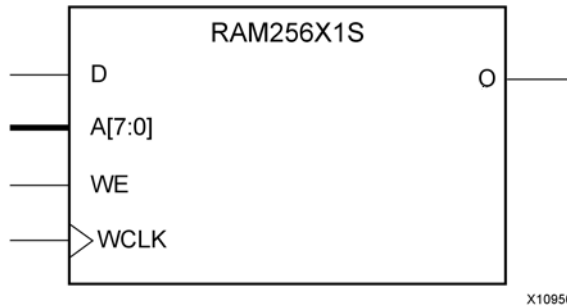


## RAM256X1S

Primitive: 256-Deep by 1-Wide Random Access Memory (Select RAM)

PRIMITIVE\_GROUP: CLB

PRIMITIVE\_SUBGROUP: LUTRAM



### 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 CLB.

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 cell 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
IP Catalog	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.
- Connect the WE clock enable pin to the proper write enable source in the design.
- Connect the 8-bit A bus to the source for the read/write.

You can use the INIT attribute to specify the initial contents of the RAM. If left unspecified, the initial contents default to all zeros.

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	HEX	Any 256-bit value	All zeros	Specifies the initial contents of the RAM.
IS_WCLK_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the WCLK pin.

## VHDL 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
--           UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

RAM256X1S_inst : RAM256X1S
generic map (
  INIT => X"0000000000000000000000000000000000000000000000000000000000000000", -- Initial contents of RAM
  IS_WCLK_INVERTED => '0') -- Specifies active high/low WCLK
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 (Mapped to four LUT6s)
//           single-port distributed LUT RAM
//           UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

RAM256X1S #(
  .INIT(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .IS_WCLK_INVERTED(1'b0) // Specifies active high/low WCLK
) RAM256X1S_inst (
  .O(O), // Read/write port 1-bit output
  .A(A), // Read/write 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
```

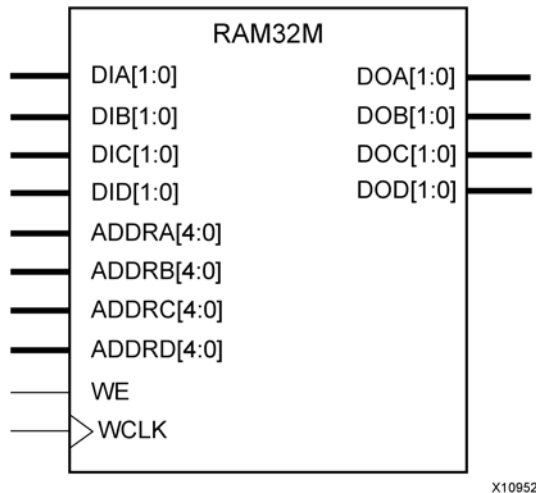
## For More Information

- See the *UltraScale Architecture Configurable Logic Block User Guide* ([UG574](#)).
- See the [UltraScale Programmable Devices Documentation](#).

# RAM32M

Primitive: 32-Deep by 8-bit Wide Multi Port Random Access Memory (Select RAM)

PRIMITIVE\_GROUP: CLB  
 PRIMITIVE\_SUBGROUP: LUTRAM



## 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, which 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.
- If ADDRA, ADDR B, and ADDR C are tied to the same address, the RAM becomes a 32x6 simple dual port RAM.
- If ADDR D is tied to ADDRA, ADDR B, and ADDR C, 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 ADDR B
DOC	Output	2	Read port data outputs addressed by ADDRC
DOD	Output	2	Read/Write port data outputs addressed by ADDR D
DIA	Input	2	Write data inputs addressed by ADDR D (read output is addressed by ADDRA)
DIB	Input	2	Write data inputs addressed by ADDR D (read output is addressed by ADDR B)
DIC	Input	2	Write data inputs addressed by ADDR D (read output is addressed by ADDRC)
DID	Input	2	Write data inputs addressed by ADDR D
ADDRA	Input	5	Read address bus A
ADDR B	Input	5	Read address bus B
ADDRC	Input	5	Read address bus C
ADDR D	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
IP Catalog	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. You should instantiate this component if you have a need to implicitly specify the RAM function, or if you need to manually place or relationally place the component.

If synchronous read capability is desired, the outputs can be connected to an FDRE/FDSE (FDCE/FDPE if asynchronous reset is needed) 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 during implementation and set as the IS\_WCLK\_INVERTED attribute giving the ability to write to the RAM on falling clock edges.

If instantiated, the following connections should be made to this component:

- Connect the WCLK input to the desired clock source
- Connect the DIA, DIB, DIC, and DID inputs to the data source to be stored
- Connect the DOA, DOB, DOC, and DOD outputs to an FDCE D input or other appropriate data destination, or leave unconnected if not used
- Connect the WE clock enable pin to the proper write enable source in the design
- Connect the ADDRD bus to the source for the read/write addressing
- Connect the ADDRA, ADDR B, and ADDR C buses to the appropriate read address connections

The optional INIT\_A, INIT\_B, INIT\_C and INIT\_D attributes let you specify the initial memory contents of each port using a 64-bit hexadecimal value. 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 ADDR C 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 be all zeros.

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT_A	HEX	Any 64-bit value	All zeros	Specifies the initial contents of the RAM on port A.
INIT_B	HEX	Any 64-bit value	All zeros	Specifies the initial contents of the RAM on port B.
INIT_C	HEX	Any 64-bit value	All zeros	Specifies the initial contents of the RAM on port C.
INIT_D	HEX	Any 64-bit value	All zeros	Specifies the initial contents of the RAM on port D.
IS_WCLK_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the WCLK pin.

## VHDL 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
-- UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

RAM32M_inst : RAM32M
generic map (
    INIT_A => X"0000000000000000", -- Initial contents of A port
    INIT_B => X"0000000000000000", -- Initial contents of B port
    INIT_C => X"0000000000000000", -- Initial contents of C port
    INIT_D => X"0000000000000000", -- Initial contents of D port
    IS_WCLK_INVERTED => '0') -- Specifies active high/low WCLK
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 => ADDR_A, -- Read port A 5-bit address input
ADDRB => ADDR_B, -- Read port B 5-bit address input
ADDR_C => ADDR_C, -- Read port C 5-bit address input
ADDRD => ADDR_D, -- Read/Write port D 5-bit address input
DIA => DIA, -- RAM 2-bit data write input addressed by ADDR_D,
-- read addressed by ADDR_A
DIB => DIB, -- RAM 2-bit data write input addressed by ADDR_D,
-- read addressed by ADDR_B
DIC => DIC, -- RAM 2-bit data write input addressed by ADDR_D,
-- read addressed by ADDR_C
DID => DID, -- RAM 2-bit data write input addressed by ADDR_D,
-- read addressed by ADDR_D
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 (Mapped to four LUT6s)
// UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

RAM32M #(
    .INIT_A(64'h0000000000000000), // Initial contents of A Port
    .INIT_B(64'h0000000000000000), // Initial contents of B Port
    .INIT_C(64'h0000000000000000), // Initial contents of C Port
    .INIT_D(64'h0000000000000000), // Initial contents of D Port
    .IS_WCLK_INVERTED(1'b0) // Specifies active high/low WCLK
) 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), // Read/write port D 2-bit output
    .ADDR_A(ADDR_A), // Read port A 5-bit address input
    .ADDR_B(ADDR_B), // Read port B 5-bit address input
    .ADDR_C(ADDR_C), // Read port C 5-bit address input
    .ADDR_D(ADDR_D), // Read/write port D 5-bit address input
    .DIA(DIA), // RAM 2-bit data write input addressed by ADDR_D,
// read addressed by ADDR_A
    .DIB(DIB), // RAM 2-bit data write input addressed by ADDR_D,
// read addressed by ADDR_B
    .DIC(DIC), // RAM 2-bit data write input addressed by ADDR_D,
// read addressed by ADDR_C
    .DID(DID), // RAM 2-bit data write input addressed by ADDR_D,
// read addressed by ADDR_D
    .WCLK(WCLK), // Write clock input
    .WE(WE) // Write enable input
);
// End of RAM32M_inst instantiation
    
```

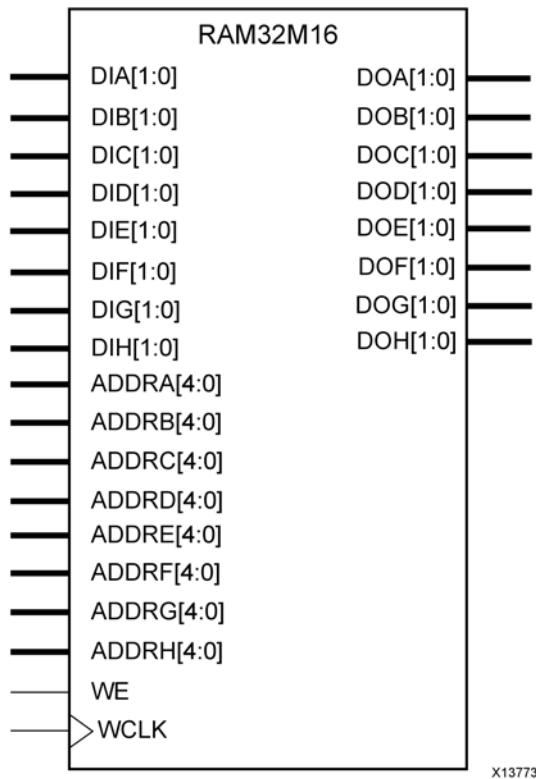
## For More Information

- See the *UltraScale Architecture Migration: Methodology Guide* ([UG1026](#)).
- See the [UltraScale Programmable Devices Documentation](#).

# RAM32M16

Primitive: 32-Deep by 16-bit Wide Multi Port Random Access Memory (Select RAM)

PRIMITIVE\_GROUP: CLB  
 PRIMITIVE\_SUBGROUP: LUTRAM



## Introduction

This design element is a 32-bit deep by 16-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. This component is implemented in a single CLB and consists of one 16-bit write, 2-bit read port and seven separate 2-bit read ports from the same memory, which allows for dual byte-wide write and independent 2-bit read access RAM.

- If the DIA through DIH inputs are all tied to the same data inputs, the RAM can become a 1 read/write port, 7 independent read port, 32x2 eight port memory.
- If DIH is grounded, DOH is not used.
- If ADDRA through ADDRG are tied to the same address, the RAM becomes a 32x14 simple dual port RAM.
- If ADDRA through ADDRH are tied together, the RAM becomes a 32x16 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 ADDR B
DOC	Output	2	Read port data outputs addressed by ADDRC
DOD	Output	2	Read port data outputs addressed by ADDR D
DOE	Output	2	Read port data outputs addressed by ADDRE
DOF	Output	2	Read port data outputs addressed by ADDR F
DOG	Output	2	Read port data outputs addressed by ADDR G
DOH	Output	2	Read/Write port data outputs addressed by ADDR H
DIA	Input	2	Data write input addressed by ADDR H (read output is addressed by ADDRA)
DIB	Input	2	Data write input addressed by ADDR H (read output is addressed by ADDR B)
DIC	Input	2	Data write input addressed by ADDR H (read output is addressed by ADDRC)
DID	Input	2	Data write input addressed by ADDR H (read output is addressed by ADDR D)
DIE	Input	2	Data write input addressed by ADDR H (read output is addressed by ADDRE)
DIF	Input	2	Data write input addressed by ADDR H (read output is addressed by ADDR F)
DIG	Input	2	Data write input addressed by ADDR H (read output is addressed by ADDR G)
DIH	Input	2	RAM 2-bit data write input addressed by ADDR H (read output is addressed by ADDR H)
ADDRA	Input	5	Read port A address input
ADDRB	Input	5	Read port B address input
ADDRC	Input	5	Read port C address input
ADDRD	Input	5	Read port D address input
ADDRE	Input	5	Read port E address input
ADDRF	Input	5	Read port F address input
ADDRG	Input	5	Read port G address input
ADDRH	Input	5	Read/write port H address input
WE	Input	1	Write Enable
WCLK	Input	1	Write clock (reads are asynchronous)

## Design Entry Method

Instantiation	Yes
Inference	Recommended
IP Catalog	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. You should instantiate this component if you have a need to implicitly specify the RAM function, or if you need to manually place or relationally place the component.

If synchronous read capability is desired, the outputs can be connected to an FDRE/FDSE (FDCE/FDPE if asynchronous reset is needed) 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 during implementation and set as the IS\_WCLK\_INVERTED attribute giving the ability to write to the RAM on falling clock edges.

If instantiated, the following connections should be made to this component:

- Connect the WCLK input to the desired clock source
- Connect the DIA–DIH inputs to the data source to be stored
- Connect the DOA–DOH outputs to an FD\* D input or other appropriate data destination, or leave unconnected if not used
- Connect the WE clock enable pin to the proper write enable source in the design
- Connect the ADDR bus to the source for the read/write addressing
- Connect the ADDRA–ADDRG buses to the appropriate read address connections

The optional INIT\_A–INIT\_H attributes let you specify the initial memory contents of each port using a 64-bit hexadecimal value. 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 ADDR\_C 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 be all zeros.

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT_A	HEX	Any 128-bit value	All zeros	Specifies the initial contents of the RAM on port A.
INIT_B	HEX	Any 128-bit value	All zeros	Specifies the initial contents of the RAM on port B.
INIT_C	HEX	Any 128-bit value	All zeros	Specifies the initial contents of the RAM on port C.
INIT_D	HEX	Any 128-bit value	All zeros	Specifies the initial contents of the RAM on port D.
INIT_E	HEX	Any 128-bit value	All zeros	Specifies the initial contents of the RAM on port E.

Attribute	Type	Allowed Values	Default	Description
INIT_F	HEX	Any 128-bit value	All zeros	Specifies the initial contents of the RAM on port F.
INIT_G	HEX	Any 128-bit value	All zeros	Specifies the initial contents of the RAM on port G.
INIT_H	HEX	Any 128-bit value	All zeros	Specifies the initial contents of the RAM on port H.
IS_WCLK_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the WCLK pin.

## VHDL Instantiation Template

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

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

```
-- RAM32M16: 32-deep by 16-wide Multi Port LUT RAM
-- UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1
RAM32M16_inst : RAM32M16
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
    INIT_E => X"0000000000000000", -- Initial contents of E port
    INIT_F => X"0000000000000000", -- Initial contents of F port
    INIT_G => X"0000000000000000", -- Initial contents of G port
    INIT_H => X"0000000000000000", -- Initial contents of H port
    IS_WCLK_INVERTED => '0') -- Specifies active high/low WCLK
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 port D 2-bit output
    DOE => DOE, -- Read port E 2-bit output
    DOF => DOF, -- Read port F 2-bit output
    DOG => DOG, -- Read port G 2-bit output
    DOH => DOH, -- Read/write port H 2-bit output
    ADDR_A => ADDR_A, -- Read port A 5-bit address input
    ADDR_B => ADDR_B, -- Read port B 5-bit address input
    ADDR_C => ADDR_C, -- Read port C 5-bit address input
    ADDR_D => ADDR_D, -- Read port D 5-bit address input
    ADDR_E => ADDR_E, -- Read port E 5-bit address input
    ADDR_F => ADDR_F, -- Read port F 5-bit address input
    ADDR_G => ADDR_G, -- Read port G 5-bit address input
    ADDR_H => ADDR_H, -- Read/write port H 5-bit address input
    DIA => DIA, -- RAM 2-bit data write input addressed by ADDR_D,
    -- read addressed by ADDR_A
    DIB => DIB, -- RAM 2-bit data write input addressed by ADDR_D,
    -- read addressed by ADDR_B
    DIC => DIC, -- RAM 2-bit data write input addressed by ADDR_D,
    -- read addressed by ADDR_C
    DID => DID, -- RAM 2-bit data write input addressed by ADDR_D,
    -- read addressed by ADDR_D
    DIE => DIE, -- RAM 2-bit data write input addressed by ADDR_E,
    -- read addressed by ADDR_E
    DIF => DIF, -- RAM 2-bit data write input addressed by ADDR_F,
    -- read addressed by ADDR_F
    DIG => DIG, -- RAM 2-bit data write input addressed by ADDR_G,
    -- read addressed by ADDR_G
    DIH => DIH, -- RAM 2-bit data write input addressed by ADDR_H,
```

```

        -- read addressed by ADDRH
        WCLK => WCLK, -- Write clock input
        WE => WE      -- Write enable input
    );
    -- End of RAM32M16_inst instantiation

```

## Verilog Instantiation Template

```

// RAM32M16: 32-deep by 16-wide Multi Port LUT RAM (Mapped to eight LUT6s)
//      UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

RAM32M16 #(
    .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
    .INIT_E(64'h0000000000000000), // Initial contents of E Port
    .INIT_F(64'h0000000000000000), // Initial contents of F Port
    .INIT_G(64'h0000000000000000), // Initial contents of G Port
    .INIT_H(64'h0000000000000000), // Initial contents of H Port
    .IS_WCLK_INVERTED(1'b0)        // Specifies active high/low WCLK
) RAM32M16_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), // Read port D 2-bit output
    .DOE(DOE), // Read port E 2-bit output
    .DOF(DOF), // Read port F 2-bit output
    .DOG(DOG), // Read port G 2-bit output
    .DOH(DOH), // Read/write port H 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 port D 5-bit address input
    .ADDRE(ADDRE), // Read port E 5-bit address input
    .ADDRF(ADDRF), // Read port F 5-bit address input
    .ADDRG(ADDRG), // Read port G 5-bit address input
    .ADDRH(ADDRH), // Read/write port H 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 ADDR B
    .DIC(DIC), // RAM 2-bit data write input addressed by ADDRD,
               // read addressed by ADDR C
    .DID(DID), // RAM 2-bit data write input addressed by ADDRD,
               // read addressed by ADDR D
    .DIE(DIE), // RAM 2-bit data write input addressed by ADDRD,
               // read addressed by ADDR E
    .DIF(DIF), // RAM 2-bit data write input addressed by ADDRD,
               // read addressed by ADDR F
    .DIG(DIG), // RAM 2-bit data write input addressed by ADDRD,
               // read addressed by ADDR G
    .DIH(DIH), // RAM 2-bit data write input addressed by ADDRD,
               // read addressed by ADDR H
    .WCLK(WCLK), // Write clock input
    .WE(WE)      // Write enable input
);

// End of RAM32M16_inst instantiation

```

## For More Information

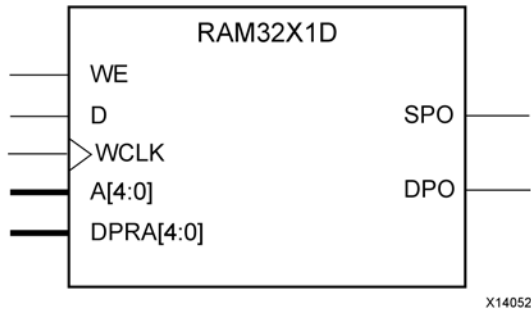
- See the *UltraScale Architecture Configurable Logic Block User Guide* ([UG574](#)).
- See the [UltraScale Programmable Devices Documentation](#).

## RAM32X1D

Primitive: 32-Deep by 1-Wide Static Dual Port Synchronous RAM

PRIMITIVE\_GROUP: CLB

PRIMITIVE\_SUBGROUP: LUTRAM



### Introduction

This design element is a 32-bit deep by 1-bit wide 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 memory cell 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.

You can use the INIT attribute to specify the initial contents of the RAM. If left unspecified, the initial contents default to all zeros.

### 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
IP Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	HEX	Any 32-bit value	All zeros	Specifies the initial contents of the RAM.
IS_WCLK_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the WCLK pin.

## VHDL 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
--           UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

RAM32X1D_inst : RAM32X1D
generic map (
    INIT => X"00000000", -- Initial contents of RAM
    IS_WCLK_INVERTED => '0') -- Specifies active high/low WCLK
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 (Mapped to two LUT6s)
//          UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

RAM32X1D #(
    .INIT(32'h00000000), // Initial contents of RAM
    .IS_WCLK_INVERTED(1'b0) // Specifies active high/low WCLK
) 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
```

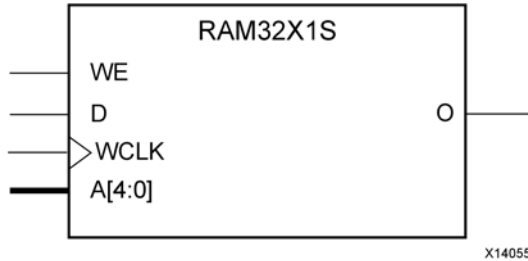
## For More Information

- See the *UltraScale Architecture Migration: Methodology Guide* ([UG1026](#)).
- See the [UltraScale Programmable Devices Documentation](#).

## RAM32X1S

Primitive: 32-Deep by 1-Wide Static Synchronous RAM

PRIMITIVE\_GROUP: CLB  
PRIMITIVE\_SUBGROUP: LUTRAM



### Introduction

This design element is a 32-bit deep by 1-bit wide 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 memory cell 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 use the INIT attribute to specify the initial contents of the RAM. If left unspecified, the initial contents default to all zeros.

### 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
IP Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	HEX	Any 32-bit value	All zeros	Specifies the initial contents of the RAM.
IS_WCLK_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the WCLK pin.

## VHDL 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
-- UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

RAM32X1S_inst : RAM32X1S
generic map (
    INIT => X"00000000", -- Initial contents of RAM
    IS_WCLK_INVERTED => '0') -- Specifies active high/low WCLK
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 (Mapped to a LUT6)
// UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

RAM32X1S #(
    .INIT(32'h00000000), // Initial contents of RAM
    .IS_WCLK_INVERTED(1'b0) // Specifies active high/low WCLK
) 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

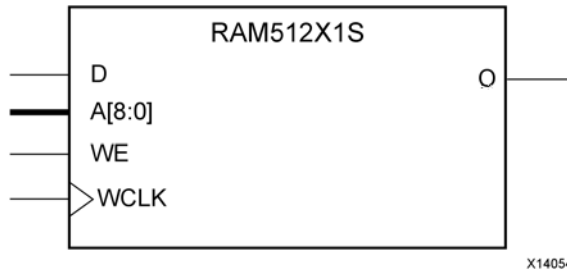
- See the *UltraScale Architecture Configurable Logic Block User Guide* ([UG574](#)).
- See the [UltraScale Programmable Devices Documentation](#).

## RAM512X1S

Primitive: 512-Deep by 1-Wide Random Access Memory (Select RAM)

PRIMITIVE\_GROUP: CLB

PRIMITIVE\_SUBGROUP: LUTRAM



### Introduction

This design element is a 512-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 RAM512X1S 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 cell 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	9	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
IP Catalog	No



```
.D(D)          // RAM data input  
);  
// End of RAM512X1S_inst instantiation
```

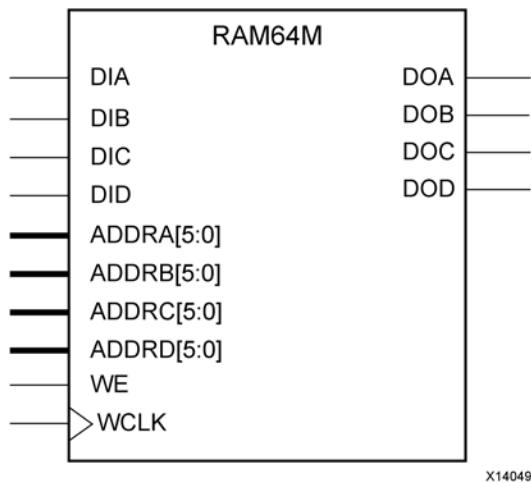
## For More Information

- See the *UltraScale Architecture Configurable Logic Block User Guide* ([UG574](#)).
- See the [UltraScale Programmable Devices Documentation](#).

## RAM64M

Primitive: 64-Deep by 4-bit Wide Multi Port Random Access Memory (Select RAM)

PRIMITIVE\_GROUP: CLB  
PRIMITIVE\_SUBGROUP: LUTRAM



### 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.
- If ADDRA, ADDR B, and ADDR C are tied to the same address, the RAM becomes a 64x3 simple dual port RAM.
- If ADDR D is tied to ADDRA, ADDR B, and ADDR C, 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 ADDR B
DOC	Output	1	Read port data outputs addressed by ADDRC
DOD	Output	1	Read/Write port data outputs addressed by ADDR D
DIA	Input	1	Write data inputs addressed by ADDR D (read output is addressed by ADDRA)
DIB	Input	1	Write data inputs addressed by ADDR D (read output is addressed by ADDR B)
DIC	Input	1	Write data inputs addressed by ADDR D (read output is addressed by ADDRC)
DID	Input	1	Write data inputs addressed by ADDR D
ADDRA	Input	6	Read address bus A
ADDR B	Input	6	Read address bus B
ADDRC	Input	6	Read address bus C
ADDR D	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
IP Catalog	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 this component if you have a need to implicitly specify the RAM function, or if you need to manually place or relationally place the component.

If synchronous read capability is desired, the outputs can be connected to an FDRE/FDSE (FDCE/FDPE if asynchronous reset is needed) 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 during implementation and set as the IS\_WCLK\_INVERTED attribute giving the ability to write to the RAM on falling clock edges.

If instantiated, the following connections should be made to this component:

- Connect the WCLK input to the desired clock source, the DIA, DIB, DIC
- Connect the DIA, DIB, DIC, and DID inputs to the data source to be stored
- Connect the DOA, DOB, DOC, and DOD outputs to an FDCE D input or other appropriate data destination, or leave unconnected if not used
- Connect the WE clock enable pin to the proper write enable source in the design
- Connect the ADDRD bus to the source for the read/write addressing
- Connect the ADDRA, ADDRB, and ADDRC buses to the appropriate read address connections

The optional INIT\_A, INIT\_B, INIT\_C and INIT\_D attributes let you specify the initial memory contents of each port using a 64-bit hexadecimal value. 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	Type	Allowed Values	Default	Description
INIT_A	HEX	Any 64-bit value	All zeros	Specifies the initial contents of the RAM on port A.
INIT_B	HEX	Any 64-bit value	All zeros	Specifies the initial contents of the RAM on port B.
INIT_C	HEX	Any 64-bit value	All zeros	Specifies the initial contents of the RAM on port C.
INIT_D	HEX	Any 64-bit value	All zeros	Specifies the initial contents of the RAM on port D.
IS_WCLK_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the WCLK pin.

## VHDL 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
-- UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

RAM64M_inst : RAM64M
generic map (
    INIT_A => X"0000000000000000", -- Initial contents of A port
    INIT_B => X"0000000000000000", -- Initial contents of B port
    INIT_C => X"0000000000000000", -- Initial contents of C port
    INIT_D => X"0000000000000000", -- Initial contents of D port
    IS_WCLK_INVERTED => '0') -- Specifies active high/low WCLK
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 => ADDR A, -- Read port A 6-bit address input
ADDRB => ADDR B, -- Read port B 6-bit address input
ADDRC => ADDR C, -- Read port C 6-bit address input
ADDRD => ADDR D, -- Read/Write port D 6-bit address input
DIA => DIA, -- RAM 1-bit data write input addressed by ADDR D,
-- read addressed by ADDR A
DIB => DIB, -- RAM 1-bit data write input addressed by ADDR D,
-- read addressed by ADDR B
DIC => DIC, -- RAM 1-bit data write input addressed by ADDR D,
-- read addressed by ADDR C
DID => DID, -- RAM 1-bit data write input addressed by ADDR D,
-- read addressed by ADDR D
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 (Mapped to four LUT6s)
// UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

RAM64M #(
    .INIT_A(64'h0000000000000000), // Initial contents of A Port
    .INIT_B(64'h0000000000000000), // Initial contents of B Port
    .INIT_C(64'h0000000000000000), // Initial contents of C Port
    .INIT_D(64'h0000000000000000), // Initial contents of D Port
    .IS_WCLK_INVERTED(1'b0) // Specifies active high/low WCLK
) 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), // Read/write port D 1-bit output
    .DIA(DIA), // RAM 1-bit data write input addressed by ADDR D,
// read addressed by ADDR A
    .DIB(DIB), // RAM 1-bit data write input addressed by ADDR D,
// read addressed by ADDR B
    .DIC(DIC), // RAM 1-bit data write input addressed by ADDR D,
// read addressed by ADDR C
    .DID(DID), // RAM 1-bit data write input addressed by ADDR D,
// read addressed by ADDR D
    .ADDRA(ADDR A), // Read port A 6-bit address input
    .ADDRB(ADDR B), // Read port B 6-bit address input
    .ADDRC(ADDR C), // Read port C 6-bit address input
    .ADDRD(ADDR D), // Read/write port D 6-bit address input
    .WE(WE), // Write enable input
    .WCLK(WCLK) // Write clock input
);

// End of RAM64M_inst instantiation
    
```

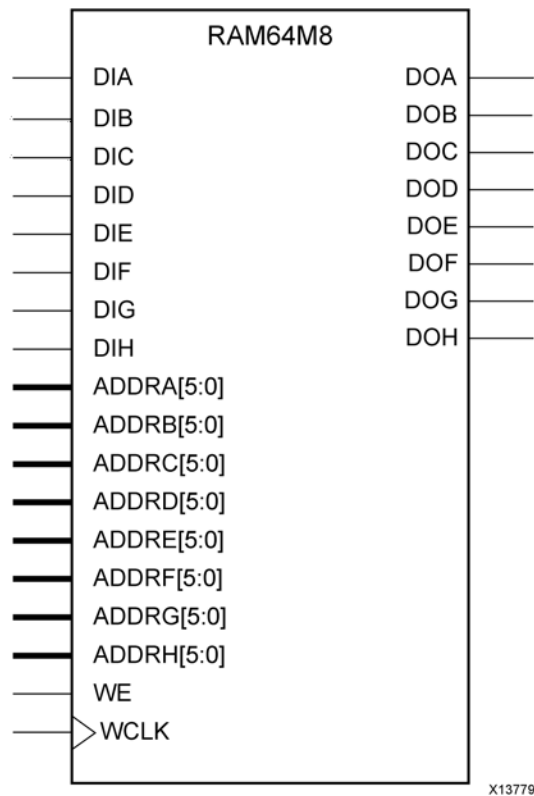
## For More Information

- See the *UltraScale Architecture Migration: Methodology Guide* ([UG1026](#)).
- See the [UltraScale Programmable Devices Documentation](#).

## RAM64M8

Primitive: 64-Deep by 8-bit Wide Multi Port Random Access Memory (Select RAM)

PRIMITIVE\_GROUP: CLB  
PRIMITIVE\_SUBGROUP: LUTRAM



### Introduction

This design element is a 64-bit deep by 8-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. This component is implemented in a single CLB and consists of one 8-bit write, 1-bit read port, and seven separate 1-bit read ports from the same memory allowing for byte-wide write and independent bit read access RAM.

- If the 7 inputs are all tied to the same data inputs, the RAM can become a 1 read/write port, 7 independent read port 64x1 octal port memory.
- If DIH is grounded, DOH is not used.
- If ADDRA through ADDR G are tied to the same address, the RAM becomes a 64x14 simple dual port RAM.
- If ADDRA through ADDR H are tied together, the RAM becomes a 64x16 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 ADDR B
DOC	Output	1	Read port data outputs addressed by ADDR C
DOD	Output	1	Read port data outputs addressed by ADDR D
DOE	Output	1	Read port data outputs addressed by ADDR E
DOF	Output	1	Read port data outputs addressed by ADDR F
DOG	Output	1	Read port data outputs addressed by ADDR G
DOH	Output	1	Read/Write port data outputs addressed by ADDR H
DIA	Input	1	Data write input addressed by ADDR H (read output is addressed by ADDRA)
DIB	Input	1	Data write input addressed by ADDR H (read output is addressed by ADDR B)
DIC	Input	1	Data write input addressed by ADDR H (read output is addressed by ADDR C)
DID	Input	1	Data write input addressed by ADDR H (read output is addressed by ADDR D)
DIE	Input	1	Data write input addressed by ADDR H (read output is addressed by ADDR E)
DIF	Input	1	Data write input addressed by ADDR H (read output is addressed by ADDR F)
DIG	Input	1	Data write input addressed by ADDR H (read output is addressed by ADDR G)
DIH	Input	1	RAM 2-bit data write input addressed by ADDR H (read output is addressed by ADDR H)
ADDRA	Input	6	Read port A address input
ADDR B	Input	6	Read port B address input
ADDR C	Input	6	Read port C address input
ADDR D	Input	6	Read port D address input
ADDR E	Input	6	Read port E address input
ADDR F	Input	6	Read port F address input
ADDR G	Input	6	Read port G address input
ADDR H	Input	6	Read/write port H address input
WE	Input	1	Write Enable
WCLK	Input	1	Write clock (reads are asynchronous)

## Design Entry Method

Instantiation	Yes
Inference	Recommended
IP Catalog	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 this component if you have a need to implicitly specify the RAM function, or if you need to manually place or relationally place the component.

If synchronous read capability is desired, the outputs can be connected to an FDRE/FDSE (FDCE/FDPE if asynchronous reset is needed) 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 during implementation and set as the IS\_WCLK\_INVERTED attribute giving the ability to write to the RAM on falling clock edges.

If instantiated, the following connections should be made to this component:

- Connect the WCLK input to the desired clock source
- Connect the DIA–DIH inputs to the data source to be stored
- Connect the DOA–DOH outputs to an FD\* D input or other appropriate data destination, or leave unconnected if not used
- Connect the WE clock enable pin to the proper write enable source in the design
- Connect the ADDRH bus to the source for the read/write addressing
- Connect the ADDRA–ADDRG buses to the appropriate read address connections

The optional INIT\_A–INIT\_H attributes let you specify the initial memory contents of each port using a 64-bit hexadecimal value. The INIT value correlates to the RAM addressing by the following equation:  $ADDRy[z] = INIT\_y[z]$ . For instance, if the RAM ADDR\_C 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	Type	Allowed Values	Default	Description
INIT_A	HEX	Any 128-bit value	All zeros	Specifies the initial contents of the RAM on port A.
INIT_B	HEX	Any 128-bit value	All zeros	Specifies the initial contents of the RAM on port B.
INIT_C	HEX	Any 128-bit value	All zeros	Specifies the initial contents of the RAM on port C.
INIT_D	HEX	Any 128-bit value	All zeros	Specifies the initial contents of the RAM on port D.
INIT_E	HEX	Any 128-bit value	All zeros	Specifies the initial contents of the RAM on port E.

Attribute	Type	Allowed Values	Default	Description
INIT_F	HEX	Any 128-bit value	All zeros	Specifies the initial contents of the RAM on port F.
INIT_G	HEX	Any 128-bit value	All zeros	Specifies the initial contents of the RAM on port G.
INIT_H	HEX	Any 128-bit value	All zeros	Specifies the initial contents of the RAM on port H.
IS_WCLK_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the WCLK pin.

## VHDL Instantiation Template

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

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

```
-- RAM64M8: 64-deep by 8-wide Multi Port LUT RAM
-- UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1
RAM64M8_inst : RAM64M8
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
    INIT_E => X"0000000000000000", -- Initial contents of E port
    INIT_F => X"0000000000000000", -- Initial contents of F port
    INIT_G => X"0000000000000000", -- Initial contents of G port
    INIT_H => X"0000000000000000", -- Initial contents of H port
    IS_WCLK_INVERTED => '0') -- Specifies active high/low WCLK
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 port D 1-bit output
    DOE => DOE, -- Read port E 1-bit output
    DOF => DOF, -- Read port F 1-bit output
    DOG => DOG, -- Read port G 1-bit output
    DOH => DOH, -- Read/write port H 1-bit output
    ADDR_A => ADDR_A, -- Read port A 6-bit address input
    ADDR_B => ADDR_B, -- Read port B 6-bit address input
    ADDR_C => ADDR_C, -- Read port C 6-bit address input
    ADDR_D => ADDR_D, -- Read port D 6-bit address input
    ADDR_E => ADDR_E, -- Read port E 6-bit address input
    ADDR_F => ADDR_F, -- Read port F 6-bit address input
    ADDR_G => ADDR_G, -- Read port G 6-bit address input
    ADDR_H => ADDR_H, -- Read/write port H 6-bit address input
    DIA => DIA, -- RAM 1-bit data write input addressed by ADDR_D,
    -- read addressed by ADDR_A
    DIB => DIB, -- RAM 1-bit data write input addressed by ADDR_D,
    -- read addressed by ADDR_B
    DIC => DIC, -- RAM 1-bit data write input addressed by ADDR_D,
    -- read addressed by ADDR_C
    DID => DID, -- RAM 1-bit data write input addressed by ADDR_D,
    -- read addressed by ADDR_D
    DIE => DIE, -- RAM 1-bit data write input addressed by ADDR_E,
    -- read addressed by ADDR_E
    DIF => DIF, -- RAM 1-bit data write input addressed by ADDR_F,
    -- read addressed by ADDR_F
    DIG => DIG, -- RAM 1-bit data write input addressed by ADDR_G,
    -- read addressed by ADDR_G
    DIH => DIH, -- RAM 1-bit data write input addressed by ADDR_H,
```

```

        -- read addressed by ADDRH
        WCLK => WCLK, -- Write clock input
        WE => WE      -- Write enable input
    );

-- End of RAM64M8_inst instantiation
    
```

## Verilog Instantiation Template

```

// RAM64M8: 64-deep by 8-wide Multi Port LUT RAM (Mapped to eight LUT6s)
// UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

RAM64M8 #(
    .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
    .INIT_E(64'h0000000000000000), // Initial contents of E Port
    .INIT_F(64'h0000000000000000), // Initial contents of F Port
    .INIT_G(64'h0000000000000000), // Initial contents of G Port
    .INIT_H(64'h0000000000000000), // Initial contents of H Port
    .IS_WCLK_INVERTED(1'b0)        // Specifies active high/low WCLK
) RAM64M8_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), // Read port D 1-bit output
    .DOE(DOE), // Read port E 1-bit output
    .DOF(DOF), // Read port F 1-bit output
    .DOG(DOG), // Read port G 1-bit output
    .DOH(DOH), // Read/write port H 1-bit output
    .DIA(DIA), // RAM 1-bit data write input addressed by ADDR,
                // read addressed by ADDRA
    .DIB(DIB), // RAM 1-bit data write input addressed by ADDR,
                // read addressed by ADDR B
    .DIC(DIC), // RAM 1-bit data write input addressed by ADDR,
                // read addressed by ADDR C
    .DID(DID), // RAM 1-bit data write input addressed by ADDR,
                // read addressed by ADDR D
    .DIE(DIE), // RAM 1-bit data write input addressed by ADDR,
                // read addressed by ADDR E
    .DIF(DIF), // RAM 1-bit data write input addressed by ADDR,
                // read addressed by ADDR F
    .DIG(DIG), // RAM 1-bit data write input addressed by ADDR,
                // read addressed by ADDR G
    .DIH(DIH), // RAM 1-bit data write input addressed by ADDR,
                // read addressed by ADDR H
    .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 port D 6-bit address input
    .ADDRE(ADDRE), // Read port E 6-bit address input
    .ADDRF(ADDRF), // Read port F 6-bit address input
    .ADDRG(ADDRG), // Read port G 6-bit address input
    .ADDRH(ADDRH), // Read/write port H 6-bit address input
    .WE(WE), // Write enable input
    .WCLK(WCLK) // Write clock input
);

// End of RAM64M8_inst instantiation
    
```

## For More Information

- See the *UltraScale Architecture Configurable Logic Block User Guide* ([UG574](#)).
- See the [UltraScale Programmable Devices Documentation](#).

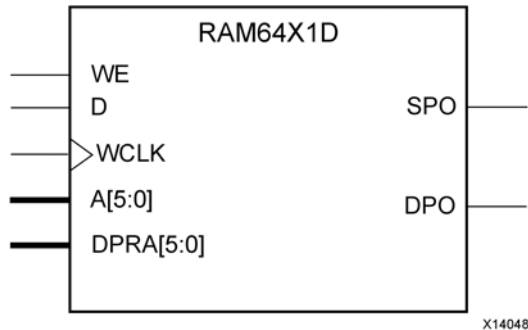


## RAM64X1D

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

PRIMITIVE\_GROUP: CLB

PRIMITIVE\_SUBGROUP: LUTRAM



### Introduction

This design element is a 64-bit deep by 1-bit wide 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 memory cell 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. The write process is not affected by the address on the read address port.

You can use the INIT attribute to specify the initial contents of the RAM. If left unspecified, the initial contents default to all zeros.

### Logic Table

Inputs			Outputs	
WE (mode)	WCLK	D	SPO	DPO
0 (read)	X	X	data_a	data_d
1 (read)	0	X	data_a	data_d
1 (read)	1	X	data_a	data_d
1 (write)	↑	D	D	data_d

Inputs			Outputs	
WE (mode)	WCLK	D	SPO	DPO
1 (read)	↓	X	data_a	data_d
data_a = memory cell addressed by bits A5:A0				
data_d = memory cell addressed by bits DPRA5:DPRA0				

## Design Entry Method

Instantiation	Yes
Inference	Recommended
IP Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	HEX	Any 64-bit value	All zeros	Specifies the initial contents of the RAM.
IS_WCLK_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the WCLK pin.

## VHDL 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 positive edge write, asynchronous read
--          dual-port distributed RAM
--          UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

RAM64X1D_inst : RAM64X1D
generic map (
    INIT => X"0000000000000000", -- Initial contents of RAM
    IS_WCLK_INVERTED => '0') -- Specifies active high/low WCLK
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_inst instantiation
```

## Verilog Instantiation Template

```
// RAM64X1D: 64 x 1 positive edge write, asynchronous read dual-port
//          distributed RAM (Mapped to two LUT6s)
//          UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

RAM64X1D #(
    .INIT(64'h0000000000000000), // Initial contents of RAM
    .IS_WCLK_INVERTED(1'b0)     // Specifies active high/low WCLK
) 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
```

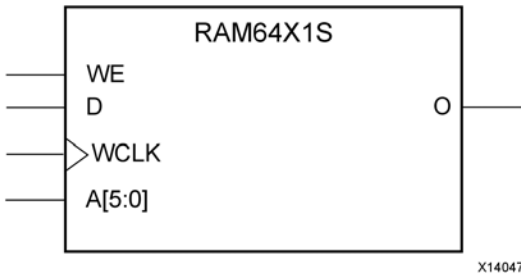
## For More Information

- See the *UltraScale Architecture Configurable Logic Block User Guide* ([UG574](#)).
- See the [UltraScale Programmable Devices Documentation](#).

# RAM64X1S

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

PRIMITIVE\_GROUP: CLB  
 PRIMITIVE\_SUBGROUP: LUTRAM



## Introduction

This design element is a 64-bit deep by 1-bit wide 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 memory cell 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 memory cell defined by the values on the address pins.

You can use the INIT attribute to specify the initial contents of the RAM. If left unspecified, the initial contents default to all zeros.

## 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 = memory cell addressed by bits A5:A0

## Design Entry Method

Instantiation	Yes
Inference	Recommended
IP Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	HEX	Any 64-bit value	All zeros	Specifies the initial contents of the RAM.
IS_WCLK_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the WCLK pin.

## VHDL 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
--           UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

RAM64X1S_inst : RAM64X1S
generic map (
    INIT => X"0000000000000000", -- Initial contents of RAM
    IS_WCLK_INVERTED => '0') -- Specifies active high/low WCLK
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 (Mapped to a LUT6)
//           UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

RAM64X1S #(
    .INIT(64'h0000000000000000), // Initial contents of RAM
    .IS_WCLK_INVERTED(1'b0)     // Specifies active high/low WCLK
) 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
```

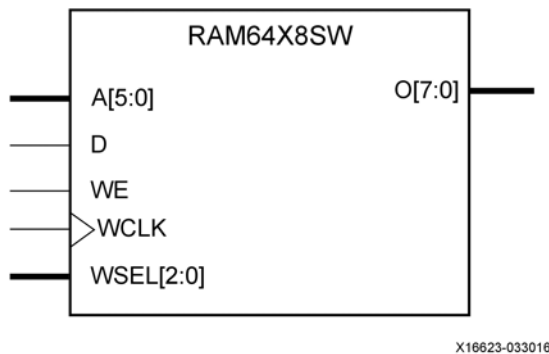
## For More Information

- See the *UltraScale Architecture Configurable Logic Block User Guide* ([UG574](#)).
- See the [UltraScale Programmable Devices Documentation](#).

## RAM64X8SW

Primitive: 64-Deep by 8-bit Wide Random Access Memory with Single-Bit Write (Select RAM)

PRIMITIVE\_GROUP: CLB  
PRIMITIVE\_SUBGROUP: LUTRAM



### Introduction

The design element is a 64-bit deep by 8-bit wide random access memory with synchronous single-bit write, and asynchronous read capability. This RAM is implemented using LUT resources of the device (also known as Select RAM), and does not consume any of the Block RAM resources of the device. This component is implemented in a single CLB and consists of a 1-bit write, 8-bit read. The RAM64X8SW has WSEL for bit-selection, and an active-High write enable, WE, so that when the 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 selected bit element. The output O displays the contents of the 8-bit memory 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
A<5:0>	Input	6	Read/Write Address Input
D	Input	1	Write Data
O<7:0>	Output	8	Read/Write port data output
WCLK	Input	1	Write Clock (reads are asynchronous)
WE	Input	1	Write Enable
WSEL<2:0>	Input	3	Write Select (Single-Bit Select)

### Design Entry Method

Instantiation	Yes
Inference	Yes
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT_A	HEX	Any 64-bit HEX value	All zeroes	Specifies the initial contents of Bit 7 of the RAM
INIT_B	HEX	Any 64-bit HEX value	All zeroes	Specifies the initial contents of Bit 6 of the RAM
INIT_C	HEX	Any 64-bit HEX value	All zeroes	Specifies the initial contents of Bit 5 of the RAM
INIT_D	HEX	Any 64-bit HEX value	All zeroes	Specifies the initial contents of Bit 4 of the RAM
INIT_E	HEX	Any 64-bit HEX value	All zeroes	Specifies the initial contents of Bit 3 of the RAM
INIT_F	HEX	Any 64-bit HEX value	All zeroes	Specifies the initial contents of Bit 2 of the RAM
INIT_G	HEX	Any 64-bit HEX value	All zeroes	Specifies the initial contents of Bit 1 of the RAM
INIT_H	HEX	Any 64-bit HEX value	All zeroes	Specifies the initial contents of Bit 0 of the RAM
IS_WCLK_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Indicates whether the WCLK is active-high or active-low

## VHDL Instantiation Template

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAM64X8SW: 64-Deep by 8-bit Wide Random Access Memory with Single-Bit Write (Select RAM)
-- UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

RAM64X8SW_inst : RAM64X8SW
generic map (
    INIT_A => X"0000000000000000", -- Initial contents of the RAM for Bit 7
    INIT_B => X"0000000000000000", -- Initial contents of the RAM for Bit 6
    INIT_C => X"0000000000000000", -- Initial contents of the RAM for Bit 5
    INIT_D => X"0000000000000000", -- Initial contents of the RAM for Bit 4
    INIT_E => X"0000000000000000", -- Initial contents of the RAM for Bit 3
    INIT_F => X"0000000000000000", -- Initial contents of the RAM for Bit 2
    INIT_G => X"0000000000000000", -- Initial contents of the RAM for Bit 1
    INIT_H => X"0000000000000000", -- Initial contents of the RAM for Bit 0
    IS_WCLK_INVERTED => '0' -- Optional inversion for WCLK
)
port map (
    O => O, -- 8-bit data output
    A => A, -- 6-bit address input
    D => D, -- 1-bit input: Write data input
    WCLK => WCLK, -- 1-bit input: Write clock input
    WE => WE, -- 1-bit input: Write enable input
    WSEL => WSEL -- 3-bit write select
);

-- End of RAM64X8SW_inst instantiation
    
```



## Verilog Instantiation Template

```
// RAM64X8SW: 64-Deep by 8-bit Wide Random Access Memory with Single-Bit Write (Select RAM)
// UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

RAM64X8SW #(
    .INIT_A(64'h0000000000000000), // Initial contents of the RAM for Bit 7
    .INIT_B(64'h0000000000000000), // Initial contents of the RAM for Bit 6
    .INIT_C(64'h0000000000000000), // Initial contents of the RAM for Bit 5
    .INIT_D(64'h0000000000000000), // Initial contents of the RAM for Bit 4
    .INIT_E(64'h0000000000000000), // Initial contents of the RAM for Bit 3
    .INIT_F(64'h0000000000000000), // Initial contents of the RAM for Bit 2
    .INIT_G(64'h0000000000000000), // Initial contents of the RAM for Bit 1
    .INIT_H(64'h0000000000000000), // Initial contents of the RAM for Bit 0
    .IS_WCLK_INVERTED(1'b0) // Optional inversion for WCLK
)
RAM64X8SW_inst (
    .O(O), // 8-bit data output
    .A(A), // 6-bit address input
    .D(D), // 1-bit input: Write data input
    .WCLK(WCLK), // 1-bit input: Write clock input
    .WE(WE), // 1-bit input: Write enable input
    .WSEL(WSEL) // 3-bit write select
);

// End of RAM64X8SW_inst instantiation
```

## For More Information

See the [UltraScale Programmable Devices User Guides](#).

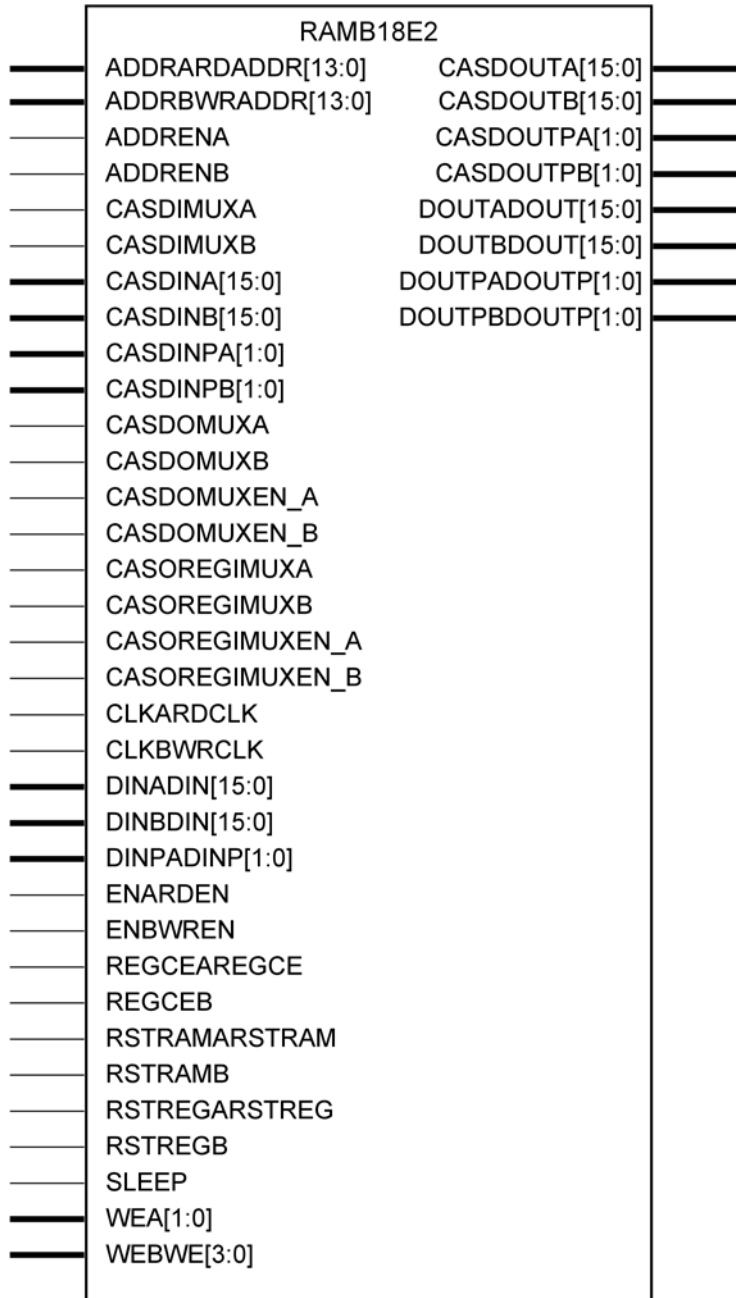
# RAMB18E2

Primitive: 18K-bit Configurable Synchronous Block RAM

PRIMITIVE\_GROUP: BLOCKRAM

PRIMITIVE\_SUBGROUP: BRAM

Families: UltraScale, UltraScale+



X14046

## Introduction

The RAMB18E2 allows access to the block RAM memory in the 18Kb configuration. This element can be configured and used as a 1-bit wide by 16K deep to an 18-bit wide by 1024-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) to the component. However, the READ and WRITE ports can operate fully independent and asynchronous to each other, accessing the same memory array. When configured in the wider data width modes, byte-enable write operations are possible, and an optional output register can be used to reduce the clock-to-out times of the RAM. This RAM also features a cascade capability, which lets you chain multiple RAMB18E2 components to form deeper and more power efficient memory configurations if desired.

## Port Descriptions

Port	Direction	Width	Function
<b>Cascade Signals:</b> Signals used when cascading more than one RAMB18E2 components			
CASDIMUXA	Input	1	Port A mux control input to select between normal data input (DINA) when Low or CASCADE data input (CASDINA) when High.
CASDIMUXB	Input	1	Port B mux control input to select between normal data input (DINB) when Low or CASCADE data input (CASDINB) when High.
CASDINA<15:0>	Input	16	Port A cascade input data
CASDINB<15:0>	Input	16	Port B cascade input data
CASDINPA<1:0>	Input	2	Port A cascade input parity data
CASDINPB<1:0>	Input	2	Port B cascade input parity data
CASDOMUXA	Input	1	Port A mux control input to select between unregistered output data from BRAM or CASCADE data input (CASDINA).
CASDOMUXB	Input	1	Port B mux control input to select between unregistered output data from BRAM or CASCADE data input (CASDINB).
CASDOMUXEN_A	Input	1	Port A unregistered output data register enable.
CASDOMUXEN_B	Input	1	Port B unregistered output data register enable.
CASDOUTA<15:0>	Output	16	Port A cascade output data
CASDOUTB<15:0>	Output	16	Port B cascade output data
CASDOUTPA<1:0>	Output	2	Port A cascade output parity data
CASDOUTPB<1:0>	Output	2	Port B cascade output parity data
CASOREGIMUXA	Input	1	Port A mux control input to select between registered output data from BRAM or CASCADE data input (CASDINA).
CASOREGIMUXB	Input	1	Port B mux control input to select between registered output data from BRAM or CASCADE data input (CASDINB).
CASOREGIMUXEN_A	Input	1	Port A registered output data register enable.

Port	Direction	Width	Function
CASOREGIMUXEN_B	Input	1	Port B registered output data register enable.
<b>Port A Address/Control Signals:</b> Port A address and control (clock, reset, enables, etc.) signals.			
ADDRARDADDR<13:0>	Input	14	Port A address input bus/Read address input bus.
ADDRENA	Input	1	Active-High port A address input bus/Read address input bus.
CLKARDCLK	Input	1	Port A clock input/Read clock input.
ENARDEN	Input	1	Port A RAM enable/Read enable. For best power characteristics of this component, it is suggested to drive this pin low when ever a new read or write operation is not necessary on Port A. Significant power savings can be seen when this port is driven low.
REGCEAREGCE	Input	1	Port A output register clock enable input/Output register clock enable input (valid only when DOA_REG=1).
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 resets port A RAM output when READ_WIDTH<=18 and the entire RAM output when READ_WIDTH=36.
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 resets port A output when READ_WIDTH<=18 and the entire RAM output when READ_WIDTH=36.
WEA<1:0>	Input	2	Port A byte-wide write enable. In the case that port A is write-only and byte-write operation is not necessary, it is suggested to connect all bits high and control write operation with the ENARDEN port. Doing this improves power consumption of the BlockRAM. In wide 36-bit mode or port A is read-only or not used, tie all bits low. In other modes, the connections are dependent on WRITE_WIDTH_A setting. See the Memory Resources User Guide for WEA mapping for different port widths.
<b>Port A Data:</b> Port A data signals.			
DINADIN<15:0>	Input	16	Port A data input bus. When WRITE_WIDTH=36, DINADIN is the logical DI<15:0>.
DINPADINP<1:0>	Input	2	Port A parity data input bus/Data parity input bus addressed by WRADDR. When WRITE_WIDTH=36, DINPADINP is the logical DIP<1:0>.
DOUTADOUT<15:0>	Output	16	Port A data output bus. When READ_WIDTH=36, DOUTADOUT is the logical DO<15:0>.
DOUTPADOUTP<1:0>	Output	2	Port A parity data output bus. When READ_WIDTH=36, DOUTPADOUTP is the logical DOP<1:0>.
<b>Port B Address/Control Signals:</b> Port B address and control (clock, reset, enables, etc.).			
ADDRBWRADDR<13:0>	Input	14	Port B address input bus/Write address input bus.

Port	Direction	Width	Function
ADDRENB	Input	1	Active-High port B address input bus/Write address input bus.
CLKBWRCLK	Input	1	Port B clock input/Write clock input.
ENBWREN	Input	1	Port B RAM enable/Write enable. For best power characteristics of this component, it is suggested to drive this pin low when ever a new read or write operation is not necessary on Port B. Significant power savings can be seen when this port is driven low.
REGCEB	Input	1	Port B output register clock enable (valid only when DOB_REG=1 and READ_WIDTH<=18).
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 READ_WIDTH=36.
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 READ_WIDTH=36.
SLEEP	Input	1	Dynamic shut down power saving. If SLEEP is High, the block is in power saving mode. If SLEEP_ASYNC=FALSE, synchronous to RDCLK, otherwise, asynchronous input. Using this pin can save additional power if the RAM is not accessed for sustained amounts of time.
WEBWE<3:0>	Input	4	Port B byte-wide write enable/Write enable. In the case that port B is write-only and byte-write operation is not necessary, it is suggested to connect all bits high and control write operation with the ENBWREN port. Doing this improves power consumption of the BlockRAM. In the case that port B is read-only or not used, tie all bits low. In other modes, the connections are dependent on WRITE_WIDTH_B setting. See the Memory Resources User Guide for WEBWE mapping for different port widths.
<b>Port B Data:</b> Port B data signals.			
DINBDIN<15:0>	Input	16	Port B data input bus. When WRITE_WIDTH=36, DINBDIN is the logical DI<31:16>.
DINPBDINP<1:0>	Input	2	Port B parity data input bus/Data parity input bus addressed by WRADDR. When WRITE_WIDTH=36, DINPBDINP is the logical DIP<3:2>.
DOUDBDOUT<15:0>	Output	16	Port B data output bus. When READ_WIDTH=36, DOUDBDOUT is the logical DO<31:16>.
DOUWPBDOUTP<1:0>	Output	2	Port B parity data output bus. When READ_WIDTH=36, DOUWPBDOUTP is the logical DOP<3:2>.

## Design Entry Method

Instantiation	Yes
Inference	Recommended
IP and IP Integrator Catalog	Yes

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
<b>CASCADE_ORDER_A, CASCADE_ORDER_B:</b> Specifies the order of the cascaded BRAM. FIRST BRAM is the bottom in cascade, LAST one is on the top of the cascade, and MIDDLE is the BRAM in between bottom and top.				
CASCADE_ORDER_A	STRING	"NONE", "FIRST", "LAST", "MIDDLE"	"NONE"	Specifies the cascade order for Port A.
CASCADE_ORDER_B	STRING	"NONE", "FIRST", "LAST", "MIDDLE"	"NONE"	Specifies the cascade order for Port B.
<b>CLOCK_DOMAINS:</b> Used for Simulation to model the address collision case as well as to enable lower power operation in common clock mode. <ul style="list-style-type: none"> <li>"COMMON"= Common Clock/Single Clock</li> <li>"INDEPENDENT"= Independent Clock/Dual Clock</li> </ul>				
CLOCK_DOMAINS	STRING	"INDEPENDENT", "COMMON"	"INDEPENDENT"	Used for Simulation to model the address collision case as well as to enable lower power operation in common clock mode. <ul style="list-style-type: none"> <li>"COMMON"= Common Clock/Single Clock</li> <li>"INDEPENDENT"= Independent Clock/Dual Clock</li> </ul>
<b>Collision check:</b> Modifies the simulation behavior so that if a memory collision occurs <ul style="list-style-type: none"> <li>"ALL" = warning produced and affected outputs/memory go unknown (X)</li> <li>"WARNING_ONLY" = warning produced and affected outputs/memory retain last value</li> <li>"GENERATE_X_ONLY" = no warning and affected outputs/memory go unknown (X)</li> <li>"NONE" = no warning and affected outputs/memory retain last value</li> </ul> <p><b>NOTE:</b> Use this setting carefully. Setting it to a value other than "ALL" can mask design problems during simulation.</p>				

Attribute	Type	Allowed Values	Default	Description
SIM_COLLISION_CHECK	STRING	"ALL", "GENERATE_X_ONLY", "NONE", "WARNING_ONLY"	"ALL"	Modifies the simulation behavior so that if a memory collision occurs <ul style="list-style-type: none"> <li>"ALL" = warning produced and affected outputs/memory go unknown (X)</li> <li>"WARNING_ONLY" = warning produced and affected outputs/memory retain last value</li> <li>"GENERATE_X_ONLY" = no warning and affected outputs/memory go unknown (X)</li> <li>"NONE" = no warning and affected outputs/memory retain last value</li> </ul> <p><b>NOTE:</b> Use this setting carefully. Setting it to a value other than "ALL" can mask design problems during simulation.</p>
<p><b>DOA_REG, DOB_REG:</b> A value of 1 enables the output registers to the RAM enabling higher performance 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.</p>				
DOA_REG	DECIMAL	1, 0	1	A value of 1 enables the output registers to the RAM enabling higher performance 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.
DOB_REG	DECIMAL	1, 0	1	A value of 1 enables the output registers to the RAM enabling higher performance 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.
<p><b>ENADDRENA/ENADDRENB:</b> Specifies if Address Enable pin is enabled</p>				
ENADDRENA	STRING	"FALSE", "TRUE"	"FALSE"	Specifies if Address Enable pin is enabled
ENADDRENB	STRING	"FALSE", "TRUE"	"FALSE"	Specifies if Address Enable pin is enabled
<p><b>INIT_A, INIT_B:</b> Specifies the initial value on the port output after configuration.</p>				
INIT_A	HEX	18'h00000 to 18'h3ffff	All zeroes	Specifies the initial value on the port output after configuration.

Attribute	Type	Allowed Values	Default	Description
INIT_B	HEX	18'h00000 to 18'h3ffff	All zeroes	Specifies the initial value on the port output after configuration.
<b>INIT_00 to INIT_3F:</b> Specifies the initial contents of the 16Kb data memory array.				
INIT_00	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_0A	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_0B	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_0C	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_0D	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_0E	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_0F	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_01	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_1A	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_1B	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_1C	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_1D	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_1E	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_1F	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_02	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_2A	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_2B	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_2C	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_2D	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_2E	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_2F	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.



Attribute	Type	Allowed Values	Default	Description
INIT_03	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_3A	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_3B	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_3C	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_3D	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_3E	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_3F	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_04	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_05	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_06	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_07	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_08	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_09	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_10	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_11	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_12	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_13	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_14	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_15	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_16	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_17	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_18	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.

Attribute	Type	Allowed Values	Default	Description
INIT_19	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_20	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_21	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_22	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_23	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_24	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_25	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_26	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_27	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_28	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_29	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_30	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_31	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_32	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_33	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_34	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_35	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_36	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_37	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_38	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
INIT_39	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 16Kb data memory array.
<b>Initialization File:</b> File name of file used to specify initial RAM contents.				
INIT_FILE	STRING	String	"NONE"	File name of file used to specify initial RAM contents.

Attribute	Type	Allowed Values	Default	Description
<b>INITP_00 to INITP_07:</b> Specifies the initial contents of the 2Kb parity data memory array.				
INITP_00	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 2Kb parity data memory array.
INITP_01	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 2Kb parity data memory array.
INITP_02	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 2Kb parity data memory array.
INITP_03	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 2Kb parity data memory array.
INITP_04	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 2Kb parity data memory array.
INITP_05	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 2Kb parity data memory array.
INITP_06	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 2Kb parity data memory array.
INITP_07	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 2Kb parity data memory array.
<b>Programmable Inversion Attributes:</b> Specifies whether or not to use the optional inversions on specific pins of this component to change the active polarity of the pin function. When set to 1 on a clock pin (WRCLK or RDCLK), this components clocks on the negative edge. When set to 1 on other pins, it changes the function to behave active-Low rather than active-High. For pins that are buses, the bit-width of this attribute should match that of the bit-width of the associated pins and a binary value indicates which inverters to use and which to bypass. If an external inverter is specified on one of these associated pins, the Vivado Design Suite will automatically set this attribute during the opt_design stage so that additional logic is not necessary for changing the input polarity.				
IS_CLKARDCLK_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the CLKARDCLK pin.
IS_CLKBWRCLK_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the CLKBWRCLK pin.
IS_ENARDEN_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the ENARDEN pin.
IS_ENBWREN_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the ENBWREN pin.
IS_RSTRAMARSTRAM_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the RSTRAMARSTRAM pin.
IS_RSTRAMB_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the RSTRAMB pin.
IS_RSTREGARSTREG_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the RSTREGARSTREG pin.

Attribute	Type	Allowed Values	Default	Description
IS_RSTREGB _INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the RSTREGB pin.
<b>RDADDRCHANGE:</b> Read Address Change is a feature which will prevent memory access when the output value does not change thus saving power. Setting this attribute to TRUE enables this power-saving feature. Setting it to false improves setup time to the RAM possibly improving BRAM performance.				
RDADDRCHANGEA	STRING	"FALSE", "TRUE"	"FALSE"	Specifies whether or not to use the Read Address Change feature for port A.
RDADDRCHANGEB	STRING	"FALSE", "TRUE"	"FALSE"	Specifies whether or not to use the Read Address Change feature for port B.
<b>READ_WIDTH_A/B, WRITE_WIDTH_A/B:</b> Specifies the desired data width for a read/write on port A/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.				
READ_WIDTH_A	DECIMAL	0, 1, 2, 4, 9, 18, 36	0	Specifies the desired data width for a read/write on port A/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.
READ_WIDTH_B	DECIMAL	0, 1, 2, 4, 9, 18	0	Specifies the desired data width for a read/write on port A/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.
WRITE_WIDTH_A	DECIMAL	0, 1, 2, 4, 9, 18	0	Specifies the desired data width for a read/write on port A/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.
WRITE_WIDTH_B	DECIMAL	0, 1, 2, 4, 9, 18, 36	0	Specifies the desired data width for a read/write on port A/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.
<b>RSTREG_PRIORITY_A, RSTREG_PRIORITY_B:</b> Selects register priority for RSTREG or REGCE				
RSTREG_PRIORITY _A	STRING	"RSTREG", "REGCE"	"RSTREG"	Selects register priority for RSTREG or REGCE
RSTREG_PRIORITY _B	STRING	"RSTREG", "REGCE"	"RSTREG"	Selects register priority for RSTREG or REGCE
<b>Sleep Async:</b> Specifies whether the SLEEP pin is synchronous to the CLKARDCLK pin ("FALSE") or treated as an asynchronous pin.				
SLEEP_ASYNC	STRING	"FALSE", "TRUE"	"FALSE"	Specifies whether the SLEEP pin is synchronous to the CLKARDCLK pin ("FALSE") or treated as an asynchronous pin.

Attribute	Type	Allowed Values	Default	Description
<b>SRVAL_A, SRVAL_B:</b> Specifies the output value of the RAM upon assertion of the synchronous reset (RST/RSTREG) signal.				
SRVAL_A	HEX	18'h00000 to 18'h3ffff	All zeroes	Specifies the output value for port A.
SRVAL_B	HEX	18'h00000 to 18'h3ffff	All zeroes	Specifies the output value for port B.
<p><b>WriteMode:</b> Specifies output behavior of the port being written to.</p> <ul style="list-style-type: none"> <li>"WRITE_FIRST" = written value appears on output port of the RAM</li> <li>"READ_FIRST" = previous RAM contents for that memory location appear on the output port. Additionally, when the same clock is used for both ports, this mode allows address collision avoidance when reading and writing to the same address from different ports.</li> <li>"NO_CHANGE" = previous value on the output port remains the same. This is the lowest power mode.</li> </ul> <p>When using the RAM as a Simple Dual Port (one port is read-only and one is write-only) it is suggested to set the WRITE_MODEs to NO_CHANGE if using different clocks on both ports or if address collisions can be avoided. It is suggested to set this to READ_FIRST mode only if using the same clock on both ports and address collisions can not be avoided in the design. WRITE_FIRST mode is not suggested to be used for Simple Dual Port operation as it consumes additional power over NO_CHANGE mode with no functional difference.</p>				

Attribute	Type	Allowed Values	Default	Description
WRITE_MODE_A	STRING	"NO_CHANGE", "READ_FIRST", "WRITE_FIRST"	"NO_CHANGE"	<p>Specifies output behavior of the port being written to.</p> <ul style="list-style-type: none"> <li>"WRITE_FIRST" = written value appears on output port of the RAM</li> <li>"READ_FIRST" = previous RAM contents for that memory location appear on the output port. Additionally, when the same clock is used for both ports, this mode allows address collision avoidance when reading and writing to the same address from different ports.</li> <li>"NO_CHANGE" = previous value on the output port remains the same. This is the lowest power mode.</li> </ul> <p>When using the RAM as a Simple Dual Port (one port is read-only and one is write-only) it is suggested to set the WRITE_MODEs to NO_CHANGE if using different clocks on both ports or if address collisions can be avoided. It is suggested to set this to READ_FIRST mode only if using the same clock on both ports and address collisions can not be avoided in the design. WRITE_FIRST mode is not suggested to be used for Simple Dual Port operation as it consumes additional power over NO_CHANGE mode with no functional difference.</p>

Attribute	Type	Allowed Values	Default	Description
WRITE_MODE_B	STRING	"NO_CHANGE", "READ_FIRST", "WRITE_FIRST"	"NO_CHANGE"	<p>Specifies output behavior of the port being written to.</p> <ul style="list-style-type: none"> <li>"WRITE_FIRST" = written value appears on output port of the RAM</li> <li>"READ_FIRST" = previous RAM contents for that memory location appear on the output port. Additionally, when the same clock is used for both ports, this mode allows address collision avoidance when reading and writing to the same address from different ports.</li> <li>"NO_CHANGE" = previous value on the output port remains the same. This is the lowest power mode.</li> </ul> <p>When using the RAM as a Simple Dual Port (one port is read-only and one is write-only) it is suggested to set the WRITE_MODEs to NO_CHANGE if using different clocks on both ports or if address collisions can be avoided. It is suggested to set this to READ_FIRST mode only if using the same clock on both ports and address collisions can not be avoided in the design. WRITE_FIRST mode is not suggested to be used for Simple Dual Port operation as it consumes additional power over NO_CHANGE mode with no functional difference.</p>

## VHDL Instantiation Template

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB18E2: 18K-bit Configurable Synchronous Block RAM
--           UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

RAMB18E2_inst : RAMB18E2
generic map (
    -- CASCADE_ORDER_A, CASCADE_ORDER_B: "FIRST", "MIDDLE", "LAST", "NONE"
    CASCADE_ORDER_A => "NONE",
    CASCADE_ORDER_B => "NONE",
    -- CLOCK_DOMAINS: "COMMON", "INDEPENDENT"
    CLOCK_DOMAINS => "INDEPENDENT",

```











```

.WRITE_WIDTH_A(0), // 0-9
.WRITE_WIDTH_B(0), // 0-9
// RSTREG_PRIORITY_A, RSTREG_PRIORITY_B: Reset or enable priority ("RSTREG", "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),
// Sleep Async: Sleep function asynchronous or synchronous ("TRUE", "FALSE")
.SLEEP_ASYNC("FALSE"),
// WriteMode: "WRITE_FIRST", "NO_CHANGE", "READ_FIRST"
.WRITE_MODE_A("NO_CHANGE"),
.WRITE_MODE_B("NO_CHANGE")
)
RAMB18E2_inst (
// Cascade Signals outputs: Multi-BRAM cascade signals
.CASDOUTA(CASDOUTA), // 16-bit output: Port A cascade output data
.CASDOUTB(CASDOUTB), // 16-bit output: Port B cascade output data
.CASDOUTPA(CASDOUTPA), // 2-bit output: Port A cascade output parity data
.CASDOUTPB(CASDOUTPB), // 2-bit output: Port B cascade output parity data
// Port A Data outputs: Port A data
.DOUTADOUT(DOUTADOUT), // 16-bit output: Port A data/LSB data
.DOUTPADOUTP(DOUTPADOUTP), // 2-bit output: Port A parity/LSB parity
// Port B Data outputs: Port B data
.DOUTBDOUT(DOUTBDOUT), // 16-bit output: Port B data/MSB data
.DOUTPBDOUTP(DOUTPBDOUTP), // 2-bit output: Port B parity/MSB parity
// Cascade Signals inputs: Multi-BRAM cascade signals
.CASDIMUXA(CASDIMUXA), // 1-bit input: Port A input data (0=DINA, 1=CASDINA)
.CASDIMUXB(CASDIMUXB), // 1-bit input: Port B input data (0=DINB, 1=CASDINB)
.CASDINA(CASDINA), // 16-bit input: Port A cascade input data
.CASDINB(CASDINB), // 16-bit input: Port B cascade input data
.CASDINPA(CASDINPA), // 2-bit input: Port A cascade input parity data
.CASDINPB(CASDINPB), // 2-bit input: Port B cascade input parity data
.CASDOMUXA(CASDOMUXA), // 1-bit input: Port A unregistered data (0=BRAM data, 1=CASDINA)
.CASDOMUXB(CASDOMUXB), // 1-bit input: Port B unregistered data (0=BRAM data, 1=CASDINB)
.CASDOMUXEN_A(CASDOMUXEN_A), // 1-bit input: Port A unregistered output data enable
.CASDOMUXEN_B(CASDOMUXEN_B), // 1-bit input: Port B unregistered output data enable
.CASOREGIMUXA(CASOREGIMUXA), // 1-bit input: Port A registered data (0=BRAM data, 1=CASDINA)
.CASOREGIMUXB(CASOREGIMUXB), // 1-bit input: Port B registered data (0=BRAM data, 1=CASDINB)
.CASOREGIMUXEN_A(CASOREGIMUXEN_A), // 1-bit input: Port A registered output data enable
.CASOREGIMUXEN_B(CASOREGIMUXEN_B), // 1-bit input: Port B registered output data enable
// Port A Address/Control Signals inputs: Port A address and control signals
.ADDRDADDR(ADDRDADDR), // 14-bit input: A/Read port address
.ADDRENA(ADDRENA), // 1-bit input: Active-High A/Read port address enable
.CLKARDCLK(CLKARDCLK), // 1-bit input: A/Read port clock
.ENARDEN(ENARDEN), // 1-bit input: Port A enable/Read enable
.REGCEAREGCE(REGCEAREGCE), // 1-bit input: Port A register enable/Register enable
.RSTRAMARSTRAM(RSTRAMARSTRAM), // 1-bit input: Port A set/reset
.RSTREGARSTREG(RSTREGARSTREG), // 1-bit input: Port A register set/reset
.WEA(WEA), // 2-bit input: Port A write enable
// Port A Data inputs: Port A data
.DINADIN(DINADIN), // 16-bit input: Port A data/LSB data
.DINPADINP(DINPADINP), // 2-bit input: Port A parity/LSB parity
// Port B Address/Control Signals inputs: Port B address and control signals
.ADDRBWRADDR(ADDRBWRADDR), // 14-bit input: B/Write port address
.ADDRENB(ADDRENB), // 1-bit input: Active-High B/Write port address enable
.CLKBWRCLK(CLKBWRCLK), // 1-bit input: B/Write port clock
.ENBWREN(ENBWREN), // 1-bit input: Port B enable/Write enable
.REGCEB(REGCEB), // 1-bit input: Port B register enable
.RSTRAMB(RSTRAMB), // 1-bit input: Port B set/reset
.RSTREGB(RSTREGB), // 1-bit input: Port B register set/reset
.SLEEP(SLEEP), // 1-bit input: Sleep Mode
.WEBWE(WEBWE), // 4-bit input: Port B write enable/Write enable
// Port B Data inputs: Port B data
.DINBDIN(DINBDIN), // 16-bit input: Port B data/MSB data
.DINPBDINP(DINPBDINP), // 2-bit input: Port B parity/MSB parity
);

// End of RAMB18E2_inst instantiation

```

## For More Information

- See the *UltraScale Architecture Memory Resources User Guide* ([UG573](#)).
- See the [UltraScale Programmable Devices Documentation](#).

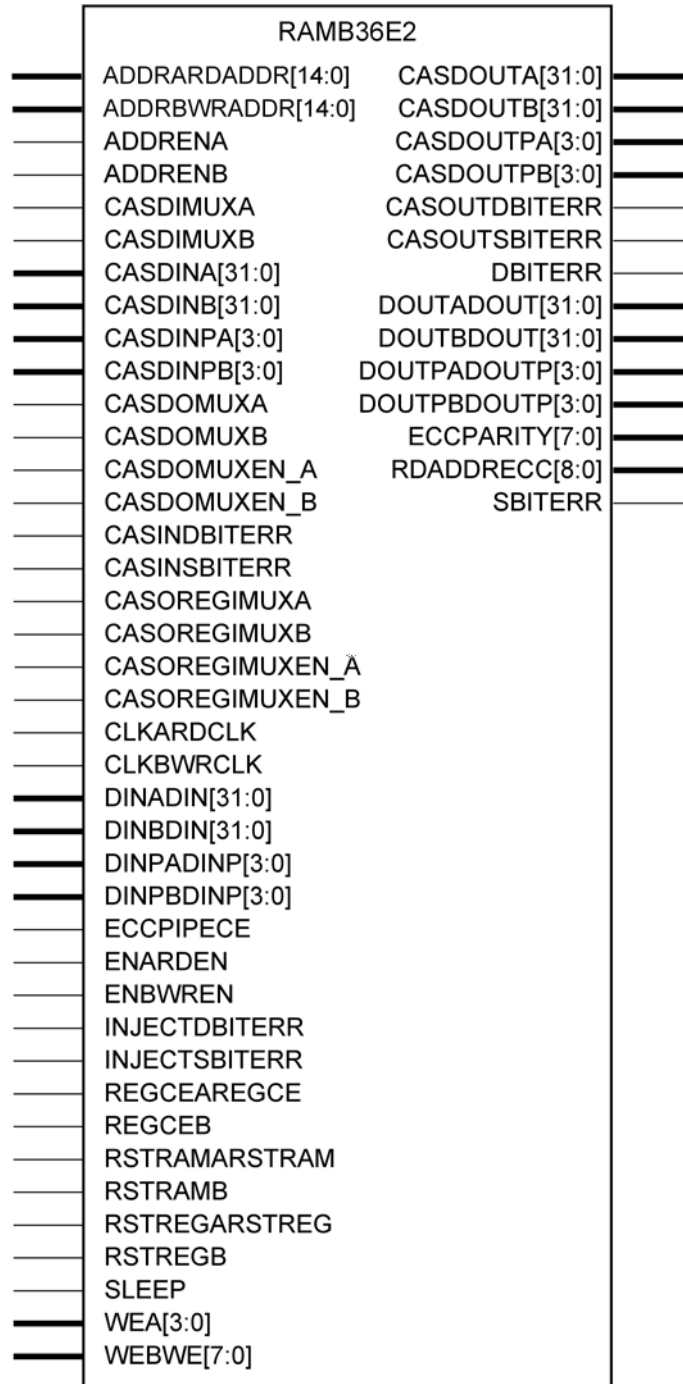
# RAMB36E2

Primitive: 36K-bit Configurable Synchronous Block RAM

PRIMITIVE\_GROUP: BLOCKRAM

PRIMITIVE\_SUBGROUP: BRAM

Families: UltraScale, UltraScale+



X14045

## Introduction

The RAMB36E2 allows access to the block RAM memory in the 36Kb configuration. This element can be configured and used as a 1-bit wide by 32K deep to an 36-bit wide by 1024-bit deep true dual port RAM. This element can also be configured as a 72-bit wide by 512 deep simple dual port RAM. Both read and write operations are fully synchronous to the supplied clock(s) to the component. However, the READ and WRITE ports can operate fully independent and asynchronous to each other, accessing the same memory array. When configured in the wider data width modes, byte-enable write operations are possible, and an optional output register can be used to reduce the clock-to-out times of the RAM. This RAM also features a cascade capability, which lets you chain multiple RAMB36E2 components to form deeper and more power efficient memory configurations if desired.

## Port Descriptions

Port	Direction	Width	Function
<b>Cascade Signals:</b> Signals used when cascading more than one RAMB36E2 components			
CASDIMUXA	Input	1	Port A mux control input to select between normal data input (DINA) when Low or CASCADE data input (CASDINA) when High.
CASDIMUXB	Input	1	Port B mux control input to select between normal data input (DINB) when Low or CASCADE data input (CASDINB) when High.
CASDINA<31:0>	Input	32	Port A cascade input data
CASDINB<31:0>	Input	32	Port B cascade input data
CASDINPA<3:0>	Input	4	Port A cascade input parity data
CASDINPB<3:0>	Input	4	Port B cascade input parity data
CASDOMUXA	Input	1	Port A mux control input to select between unregistered output data from BRAM or CASCADE data input (CASDINA).
CASDOMUXB	Input	1	Port B mux control input to select between unregistered output data from BRAM or CASCADE data input (CASDINB).
CASDOMUXEN_A	Input	1	Port A unregistered output data register enable.
CASDOMUXEN_B	Input	1	Port B unregistered output data register enable.
CASDOUTA<31:0>	Output	32	Port A cascade output data
CASDOUTB<31:0>	Output	32	Port B cascade output data
CASDOUTPA<3:0>	Output	4	Port A cascade output parity data
CASDOUTPB<3:0>	Output	4	Port B cascade output parity data
CASINDBITERR	Input	1	Cascaded double bit error (DBITERR) signal from prior BlockRAM in the cascade chain.
CASINSBITERR	Input	1	Cascaded single bit error (SBITERR) signal from prior BlockRAM in the cascade chain.
CASOREGIMUXA	Input	1	Port A mux control input to select between registered output data from BRAM or CASCADE data input (CASDINA).

Port	Direction	Width	Function
CASOREGIMUXB	Input	1	Port B mux control input to select between registered output data from BRAM or CASCADE data input (CASDINB).
CASOREGIMUXEN_A	Input	1	Port A registered output data register enable.
CASOREGIMUXEN_B	Input	1	Port B registered output data register enable.
CASOUTDBITERR	Output	1	Cascaded double bit error (DBITERR) signal to next BlockRAM in the cascade chain.
CASOUTSBITERR	Output	1	Cascaded single bit error (SBITERR) signal to next BlockRAM in the cascade chain.
<b>ECC Signals:</b> Error Correction Circuitry ports			
DBITERR	Output	1	Status output from ECC function to indicate a double bit error was detected during a read operation. EN_ECC_READ needs to be TRUE in order to use this functionality. Synchronous to RDCLK.
ECCPARITY<7:0>	Output	8	8-bit data generated by the ECC encoder used by the ECC decoder for memory error detection and correction. Applicable when EN_ECC_WRITE=1. Synchronous to WRCLK.
ECCPIPECE	Input	1	Clock enable for the ECC pipeline register.
INJECTDBITERR	Input	1	Applicable when EN_ECC_WRITE=1. Causes a double-bit error to be inserted on bits 30 and 62 of DI during a write operation. Synchronous to WRCLK.
INJECTSBITERR	Input	1	Applicable when EN_ECC_WRITE=1. Causes a single-bit error to be inserted on bit 30 of DI during a write operation.
RDADDRECC<8:0>	Output	9	Read address for ECC function.
SBITERR	Output	1	ECC output indicating that a single-bit error was detected during the read operation. Synchronous to RDCLK.
<b>Port A Address/Control Signals:</b> Port A address and control (clock, reset, enables, etc.) signals.			
ADDRARDADDR<14:0>	Input	15	Port A address input bus/Read address input bus.
ADDRENA	Input	1	Active-High port A address input bus/Read address input bus.
CLKARDCLK	Input	1	Port A clock input/Read clock input.
ENARDEN	Input	1	Port A RAM enable/Read enable. For best power characteristics of this component, it is suggested to drive this pin low when ever a new read or write operation is not necessary on Port A. Significant power savings can be seen when this port is driven low.
REGCEAREGCE	Input	1	Port A output register clock enable input/Output register clock enable input (valid only when DOA_REG=1).



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 resets port A RAM output when READ_WIDTH<=36 and the entire RAM output when READ_WIDTH=72.
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 resets port A output when READ_WIDTH<=36 and the entire RAM output when READ_WIDTH=72.
SLEEP	Input	1	Dynamic shut down power saving. If SLEEP is High, the block is in power saving mode. If SLEEP_ASYNC=FALSE, synchronous to RDCLK, otherwise, asynchronous input. Using this pin can save additional power if the RAM is not accessed for sustained amounts of time.
WEA<3:0>	Input	4	Port A byte-wide write enable. In the case that port A is write-only and byte-write operation is not necessary, it is suggested to connect all bits high and control write operation with the ENARDEN port. Doing this improves power consumption of the BlockRAM. In wide 72-bit mode or if port A is read-only or not used, tie all bits low. In other modes, the connections are dependent on WRITE_WIDTH_A setting. See the Memory Resources User Guide for WEA mapping for different port widths.
<b>Port A Data:</b> Port A data signals.			
DINADIN<31:0>	Input	32	Port A data input bus. When WRITE_WIDTH=72, DINADIN is the logical DI<31:0>.
DINPADINP<3:0>	Input	4	Port A parity data input bus/Data parity input bus addressed by WRADDR. When WRITE_WIDTH=72, DINPADINP is the logical DIP<3:0>.
DOUTADOUT<31:0>	Output	32	Port A data output bus. When READ_WIDTH=72, DOUTADOUT is the logical DO<31:0>.
DOUTPADOUTP<3:0>	Output	4	Port A parity data output bus. When READ_WIDTH=72, DOUTPADOUT is the logical DOP<3:0>.
<b>Port B Address/Control Signals:</b> Port B address and control (clock, reset, enables, etc.).			
ADDRBWRADDR<14:0>	Input	15	Port B address input bus/Write address input bus.
ADDRENB	Input	1	Active-High port B address input bus/Write address input bus.
CLKBWRCLK	Input	1	Port B clock input/Write clock input.
ENBWREN	Input	1	Port B RAM enable/Write enable. For best power characteristics of this component, it is suggested to drive this pin low when ever a new read or write operation is not necessary on Port B. Significant power savings can be seen when this port is driven low.
REGCEB	Input	1	Port B output register clock enable (valid only when DOB_REG=1 and READ_WIDTH<=36).

Port	Direction	Width	Function
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 READ_WIDTH=72.
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 READ_WIDTH=72.
WEBWE<7:0>	Input	8	Port B byte-wide write enable/Write enable. In the case that port B is write-only and byte-write operation is not necessary, it is suggested to connect all bits high and control write operation with the ENBWREN port. Doing this improves power consumption of the BlockRAM. When using ECC mode, all bits must be tied high. In the case that port B is read-only or not used, tie all bits low. In other modes, the connections are dependent on WRITE_WIDTH_B setting. See the Memory Resources User Guide for WEBWE mapping for different port widths.
<b>Port B Data:</b> Port B data signals.			
DINBDIN<31:0>	Input	32	Port B data input bus. When WRITE_WIDTH=72, DINBDIN is the logical DI<63:32>.
DINPBDINP<3:0>	Input	4	Port B parity data input bus/Data parity input bus addressed by WRADDR. When WRITE_WIDTH=72, DINPBDINP is the logical DIP<7:4>.
DOUDBDOUT<31:0>	Output	32	Port B data output bus. When READ_WIDTH=72, DOUDBDOUT is the logical DO<63:32>.
DOUDBDOUTP<3:0>	Output	4	Port B parity data output bus. When READ_WIDTH=72, DOUDBDOUTP is the logical DOP<7:4>.

## Design Entry Method

Instantiation	Yes
Inference	Recommended
IP and IP Integrator Catalog	Yes

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
<b>CASCADE_ORDER_A, CASCADE_ORDER_B:</b> Specifies the order of the cascaded BRAM. FIRST BRAM is the bottom in cascade, LAST one is on the top of the cascade, and MIDDLE is the BRAM in between bottom and top.				
CASCADE_ORDER_A	STRING	"NONE", "FIRST", "LAST", "MIDDLE"	"NONE"	Specifies the cascade order for Port A.
CASCADE_ORDER_B	STRING	"NONE", "FIRST", "LAST", "MIDDLE"	"NONE"	Specifies the cascade order for Port B.

Attribute	Type	Allowed Values	Default	Description
<p><b>CLOCK_DOMAINS:</b> Used for Simulation purpose to model the address collision case as well as to enable lower power operation in common clock mode.</p> <ul style="list-style-type: none"> <li>"COMMON"= Common Clock/Single Clock</li> <li>"INDEPENDENT"= Independent Clock/Dual Clock</li> </ul>				
CLOCK_DOMAINS	STRING	"INDEPENDENT", "COMMON"	"INDEPENDENT"	Used for Simulation purpose to model the address collision case as well as to enable lower power operation in common clock mode. <ul style="list-style-type: none"> <li>"COMMON"= Common Clock/Single Clock</li> <li>"INDEPENDENT"= Independent Clock/Dual Clock</li> </ul>
DOB_REG	DECIMAL	1, 0	1	Used for Simulation purpose to model the address collision case as well as to enable lower power operation in common clock mode. <ul style="list-style-type: none"> <li>"COMMON"= Common Clock/Single Clock</li> <li>"INDEPENDENT"= Independent Clock/Dual Clock</li> </ul>
<p><b>Collision check:</b> Modifies the simulation behavior so that if a memory collision occurs</p> <ul style="list-style-type: none"> <li>"ALL" = warning produced and affected outputs/memory go unknown (X)</li> <li>"WARNING_ONLY" = warning produced and affected outputs/memory retain last value</li> <li>"GENERATE_X_ONLY" = no warning and affected outputs/memory go unknown (X)</li> <li>"NONE" = no warning and affected outputs/memory retain last value</li> </ul> <p><b>NOTE:</b> Use this setting carefully. Setting it to a value other than "ALL" can mask design problems during simulation.</p>				

Attribute	Type	Allowed Values	Default	Description
SIM_COLLISION_CHECK	STRING	"ALL", "GENERATE_X_ONLY", "NONE", "WARNING_ONLY"	"ALL"	<p>Modifies the simulation behavior so that if a memory collision occurs</p> <ul style="list-style-type: none"> <li>"ALL" = warning produced and affected outputs/memory go unknown (X)</li> <li>"WARNING_ONLY" = warning produced and affected outputs/memory retain last value</li> <li>"GENERATE_X_ONLY" = no warning and affected outputs/memory go unknown (X)</li> <li>"NONE" = no warning and affected outputs/memory retain last value</li> </ul> <p><b>NOTE:</b> Use this setting carefully. Setting it to a value other than "ALL" can mask design problems during simulation.</p>
<p><b>DOA_REG, DOB_REG:</b> A value of 1 enables the output registers to the RAM enabling higher performance 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.</p>				
DOA_REG	DECIMAL	1, 0	1	<p>A value of 1 enables the output registers to the RAM enabling higher performance 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.</p>
<p><b>EN_ECC_PIPE:</b> Enable ECC pipeline output register stage.</p>				
EN_ECC_PIPE	STRING	"FALSE", "TRUE"	"FALSE"	<p>Enable ECC pipeline output register stage.</p>
<p><b>EN_ECC_READ:</b> Enable the ECC read decoder circuitry. Only valid when READ_WIDTH is set to 72.</p>				
EN_ECC_READ	STRING	"FALSE", "TRUE"	"FALSE"	<p>Enable the ECC read decoder circuitry. Only valid when READ_WIDTH is set to 72.</p>
<p><b>EN_ECC_WRITE:</b> Enable the ECC write encoder circuitry. When using ECC mode, byte-write capability is not supported and the WEBWE signals must all be tied high to ensure proper operation. Only valid when WRITE_WIDTH is set to 72.</p>				
EN_ECC_WRITE	STRING	"FALSE", "TRUE"	"FALSE"	<p>Enable the ECC write encoder circuitry. When using ECC mode, byte-write capability is not supported and the WEBWE signals must all be tied high to ensure proper operation. Only valid when WRITE_WIDTH is set to 72.</p>

Attribute	Type	Allowed Values	Default	Description
<b>ENADDRENA/ENADDRENB:</b> Specifies if Address Enable pin is enabled				
ENADDRENA	STRING	"FALSE", "TRUE"	"FALSE"	Specifies if Address Enable pin is enabled
ENADDRENB	STRING	"FALSE", "TRUE"	"FALSE"	Specifies if Address Enable pin is enabled
<b>INIT_A, INIT_B:</b> Specifies the initial value on the port output after configuration.				
INIT_A	HEX	Any 36-bit HEX value	All zeroes	Specifies the initial value on the port output after configuration.
INIT_B	HEX	Any 36-bit HEX value	All zeroes	Specifies the initial value on the port output after configuration.
<b>INIT_00 to INIT_7F:</b> Specifies the initial contents of the 32Kb data memory array.				
INIT_00	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_0A	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_0B	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_0C	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_0D	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_0E	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_0F	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_01	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_1A	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_1B	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_1C	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_1D	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_1E	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_1F	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_02	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_2A	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.

Attribute	Type	Allowed Values	Default	Description
INIT_2B	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_2C	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_2D	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_2E	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_2F	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_03	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_3A	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_3B	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_3C	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_3D	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_3E	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_3F	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_04	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_4A	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_4B	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_4C	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_4D	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_4E	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_4F	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_05	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_5A	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_5B	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.

Attribute	Type	Allowed Values	Default	Description
INIT_5C	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_5D	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_5E	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_5F	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_06	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_6A	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_6B	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_6C	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_6D	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_6E	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_6F	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_07	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_7A	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_7B	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_7C	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_7D	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_7E	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_7F	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_08	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_09	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_10	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_11	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.

Attribute	Type	Allowed Values	Default	Description
INIT_12	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_13	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_14	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_15	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_16	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_17	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_18	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_19	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_20	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_21	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_22	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_23	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_24	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_25	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_26	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_27	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_28	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_29	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_30	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_31	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_32	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_33	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.



Attribute	Type	Allowed Values	Default	Description
INIT_34	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_35	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_36	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_37	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_38	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_39	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_40	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_41	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_42	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_43	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_44	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_45	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_46	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_47	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_48	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_49	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_50	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_51	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_52	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_53	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_54	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_55	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.

Attribute	Type	Allowed Values	Default	Description
INIT_56	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_57	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_58	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_59	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_60	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_61	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_62	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_63	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_64	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_65	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_66	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_67	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_68	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_69	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_70	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_71	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_72	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_73	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_74	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_75	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_76	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_77	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.

Attribute	Type	Allowed Values	Default	Description
INIT_78	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
INIT_79	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 32Kb data memory array.
<b>Initialization File:</b> File name of file used to specify initial RAM contents.				
INIT_FILE	STRING	String	"NONE"	File name of file used to specify initial RAM contents.
<b>INITP_00 to INITP_0F:</b> Specifies the initial contents of the 4Kb parity data memory array.				
INITP_00	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 4Kb parity data memory array.
INITP_0A	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 4Kb parity data memory array.
INITP_0B	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 4Kb parity data memory array.
INITP_0C	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 4Kb parity data memory array.
INITP_0D	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 4Kb parity data memory array.
INITP_0E	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 4Kb parity data memory array.
INITP_0F	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 4Kb parity data memory array.
INITP_01	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 4Kb parity data memory array.
INITP_02	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 4Kb parity data memory array.
INITP_03	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 4Kb parity data memory array.
INITP_04	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 4Kb parity data memory array.
INITP_05	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 4Kb parity data memory array.
INITP_06	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 4Kb parity data memory array.
INITP_07	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 4Kb parity data memory array.
INITP_08	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 4Kb parity data memory array.
INITP_09	HEX	Any 256-bit HEX value	All zeroes	Specifies the initial contents of the 4Kb parity data memory array.

Attribute	Type	Allowed Values	Default	Description
<p><b>Programmable Inversion Attributes:</b> Specifies whether or not to use the optional inversions on specific pins of this component to change the active polarity of the pin function. When set to 1 on a clock pin (WRCLK or RDCLK), this components clocks on the negative edge. When set to 1 on other pins, it changes the function to behave active-Low rather than active-High. For pins that are buses, the bit-width of this attribute should match that of the bit-width of the associated pins and a binary value indicates which inverters to use and which to bypass. If an external inverter is specified on one of these associated pins, the Vivado Design Suite will automatically set this attribute during the opt_design stage so that additional logic is not necessary for changing the input polarity.</p>				
IS_CLKARDCLK_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversions on specific pins of this component to change the active polarity of the pin function. When set to 1 on a clock pin (WRCLK or RDCLK), this components clocks on the negative edge. When set to 1 on other pins, it changes the function to behave active-Low rather than active-High. For pins that are buses, the bit-width of this attribute should match that of the bit-width of the associated pins and a binary value indicates which inverters to use and which to bypass. If an external inverter is specified on one of these associated pins, the Vivado Design Suite will automatically set this attribute during the opt_design stage so that additional logic is not necessary for changing the input polarity.
IS_CLKBWRCLK_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversions on specific pins of this component to change the active polarity of the pin function. When set to 1 on a clock pin (WRCLK or RDCLK), this components clocks on the negative edge. When set to 1 on other pins, it changes the function to behave active-Low rather than active-High. For pins that are buses, the bit-width of this attribute should match that of the bit-width of the associated pins and a binary value indicates which inverters to use and which to bypass. If an external inverter is specified on one of these associated pins, the Vivado Design Suite will automatically set this attribute during the opt_design stage so that additional logic is not necessary for changing the input polarity.

Attribute	Type	Allowed Values	Default	Description
IS_ENARDEN_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversions on specific pins of this component to change the active polarity of the pin function. When set to 1 on a clock pin (WRCLK or RDCLK), this components clocks on the negative edge. When set to 1 on other pins, it changes the function to behave active-Low rather than active-High. For pins that are buses, the bit-width of this attribute should match that of the bit-width of the associated pins and a binary value indicates which inverters to use and which to bypass. If an external inverter is specified on one of these associated pins, the Vivado Design Suite will automatically set this attribute during the opt_design stage so that additional logic is not necessary for changing the input polarity.
IS_ENBWREN_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversions on specific pins of this component to change the active polarity of the pin function. When set to 1 on a clock pin (WRCLK or RDCLK), this components clocks on the negative edge. When set to 1 on other pins, it changes the function to behave active-Low rather than active-High. For pins that are buses, the bit-width of this attribute should match that of the bit-width of the associated pins and a binary value indicates which inverters to use and which to bypass. If an external inverter is specified on one of these associated pins, the Vivado Design Suite will automatically set this attribute during the opt_design stage so that additional logic is not necessary for changing the input polarity.

Attribute	Type	Allowed Values	Default	Description
IS_RSTRAMARSTRAM_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversions on specific pins of this component to change the active polarity of the pin function. When set to 1 on a clock pin (WRCLK or RDCLK), this components clocks on the negative edge. When set to 1 on other pins, it changes the function to behave active-Low rather than active-High. For pins that are buses, the bit-width of this attribute should match that of the bit-width of the associated pins and a binary value indicates which inverters to use and which to bypass. If an external inverter is specified on one of these associated pins, the Vivado Design Suite will automatically set this attribute during the opt_design stage so that additional logic is not necessary for changing the input polarity.
IS_RSTRAMB_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversions on specific pins of this component to change the active polarity of the pin function. When set to 1 on a clock pin (WRCLK or RDCLK), this components clocks on the negative edge. When set to 1 on other pins, it changes the function to behave active-Low rather than active-High. For pins that are buses, the bit-width of this attribute should match that of the bit-width of the associated pins and a binary value indicates which inverters to use and which to bypass. If an external inverter is specified on one of these associated pins, the Vivado Design Suite will automatically set this attribute during the opt_design stage so that additional logic is not necessary for changing the input polarity.

Attribute	Type	Allowed Values	Default	Description
IS_RSTREGARSTREG_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversions on specific pins of this component to change the active polarity of the pin function. When set to 1 on a clock pin (WRCLK or RDCLK), this components clocks on the negative edge. When set to 1 on other pins, it changes the function to behave active-Low rather than active-High. For pins that are buses, the bit-width of this attribute should match that of the bit-width of the associated pins and a binary value indicates which inverters to use and which to bypass. If an external inverter is specified on one of these associated pins, the Vivado Design Suite will automatically set this attribute during the opt_design stage so that additional logic is not necessary for changing the input polarity.
IS_RSTREGB_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversions on specific pins of this component to change the active polarity of the pin function. When set to 1 on a clock pin (WRCLK or RDCLK), this components clocks on the negative edge. When set to 1 on other pins, it changes the function to behave active-Low rather than active-High. For pins that are buses, the bit-width of this attribute should match that of the bit-width of the associated pins and a binary value indicates which inverters to use and which to bypass. If an external inverter is specified on one of these associated pins, the Vivado Design Suite will automatically set this attribute during the opt_design stage so that additional logic is not necessary for changing the input polarity.
<b>RDADDRCHANGE:</b> Read Address Change is a feature which will prevent memory access when the output value does not change thus saving power. Setting this attribute to TRUE enables this power-saving feature. Setting it to false improves setup time to the RAM possibly improving BRAM performance.				
RDADDRCHANGEA	STRING	"FALSE", "TRUE"	"FALSE"	Specifies whether or not to use the Read Address Change feature for port A.

Attribute	Type	Allowed Values	Default	Description
RDADDRCHANGE	STRING	"FALSE", "TRUE"	"FALSE"	Specifies whether or not to use the Read Address Change feature for port B.
<b>READ_WIDTH_A/B, WRITE_WIDTH_A/B:</b> Specifies the desired data width for a read/write on port A/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.				
READ_WIDTH_A	DECIMAL	0, 1, 2, 4, 9, 18, 36, 72	0	Specifies the desired data width for a read/write on port A/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.
READ_WIDTH_B	DECIMAL	0, 1, 2, 4, 9, 18, 36	0	Specifies the desired data width for a read/write on port A/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.
WRITE_WIDTH_A	DECIMAL	0, 1, 2, 4, 9, 18, 36	0	Specifies the desired data width for a read/write on port A/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.
WRITE_WIDTH_B	DECIMAL	0, 1, 2, 4, 9, 18, 36, 72	0	Specifies the desired data width for a read/write on port A/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.
<b>RSTREG_PRIORITY_A, RSTREG_PRIORITY_B:</b> Selects register priority for RSTREG or REGCE				
RSTREG_PRIORITY_A	STRING	"RSTREG", "REGCE"	"RSTREG"	Selects register priority for RSTREG or REGCE
RSTREG_PRIORITY_B	STRING	"RSTREG", "REGCE"	"RSTREG"	Selects register priority for RSTREG or REGCE
<b>Sleep Async:</b> Specifies whether the SLEEP pin is synchronous to the CLKARDCLK pin ("FALSE") or treated as an asynchronous pin.				
SLEEP_ASYNC	STRING	"FALSE", "TRUE"	"FALSE"	Specifies whether the SLEEP pin is synchronous to the CLKARDCLK pin ("FALSE") or treated as an asynchronous pin.
<b>SRVAL_A, SRVAL_B:</b> Specifies the output value of the RAM upon assertion of the synchronous reset (RST/RSTREG) signal.				
SRVAL_A	HEX	Any 36-bit HEX value	All zeroes	Specifies the output value of the RAM upon assertion of the synchronous reset (RST/RSTREG) signal.



Attribute	Type	Allowed Values	Default	Description
SRVAL_B	HEX	Any 36-bit HEX value	All zeroes	Specifies the output value of the RAM upon assertion of the synchronous reset (RST/RSTREG) signal.
<p><b>WriteMode:</b> Specifies output behavior of the port being written to.</p> <ul style="list-style-type: none"> <li>"WRITE_FIRST" = written value appears on output port of the RAM</li> <li>"READ_FIRST" = previous RAM contents for that memory location appear on the output port. Additionally, when the same clock is used for both ports, this mode allows address collision avoidance when reading and writing to the same address from different ports.</li> <li>"NO_CHANGE" = previous value on the output port remains the same. This is the lowest power mode.</li> </ul> <p>When using the RAM as a Simple Dual Port (one port is read-only and one is write-only) it is suggested to set the WRITE_MODEs to NO_CHANGE if using different clocks on both ports or if address collisions can be avoided. It is suggested to set this to READ_FIRST mode only if using the same clock on both ports and address collisions can not be avoided in the design. WRITE_FIRST mode is not suggested to be used for Simple Dual Port operation as it consumes additional power over NO_CHANGE mode with no functional difference.</p>				
WRITE_MODE_A	STRING	"NO_CHANGE", "READ_FIRST", "WRITE_FIRST"	"NO_CHANGE"	<p>Specifies output behavior of the port being written to.</p> <ul style="list-style-type: none"> <li>"WRITE_FIRST" = written value appears on output port of the RAM</li> <li>"READ_FIRST" = previous RAM contents for that memory location appear on the output port. Additionally, when the same clock is used for both ports, this mode allows address collision avoidance when reading and writing to the same address from different ports.</li> <li>"NO_CHANGE" = previous value on the output port remains the same. This is the lowest power mode.</li> </ul> <p>When using the RAM as a Simple Dual Port (one port is read-only and one is write-only) it is suggested to set the WRITE_MODEs to NO_CHANGE if using different clocks on both ports or if address collisions can be avoided. It is suggested to set this to READ_FIRST mode only if using the same clock on both ports and address collisions can not be avoided in the design. WRITE_FIRST mode is not suggested to be used for Simple Dual Port operation as it consumes additional power over NO_CHANGE mode with no functional difference.</p>

Attribute	Type	Allowed Values	Default	Description
WRITE_MODE_B	STRING	"NO_CHANGE", "READ_FIRST", "WRITE_FIRST"	"NO_CHANGE"	<p>Specifies output behavior of the port being written to.</p> <ul style="list-style-type: none"> <li>"WRITE_FIRST" = written value appears on output port of the RAM</li> <li>"READ_FIRST" = previous RAM contents for that memory location appear on the output port. Additionally, when the same clock is used for both ports, this mode allows address collision avoidance when reading and writing to the same address from different ports.</li> <li>"NO_CHANGE" = previous value on the output port remains the same. This is the lowest power mode.</li> </ul> <p>When using the RAM as a Simple Dual Port (one port is read-only and one is write-only) it is suggested to set the WRITE_MODEs to NO_CHANGE if using different clocks on both ports or if address collisions can be avoided. It is suggested to set this to READ_FIRST mode only if using the same clock on both ports and address collisions can not be avoided in the design. WRITE_FIRST mode is not suggested to be used for Simple Dual Port operation as it consumes additional power over NO_CHANGE mode with no functional difference.</p>

## VHDL Instantiation Template

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB36E2: 36K-bit Configurable Synchronous Block RAM
--           UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

RAMB36E2_inst : RAMB36E2
generic map (
    -- CASCADE_ORDER_A, CASCADE_ORDER_B: "FIRST", "MIDDLE", "LAST", "NONE"
    CASCADE_ORDER_A => "NONE",
    CASCADE_ORDER_B => "NONE",
    -- CLOCK_DOMAINS: "COMMON", "INDEPENDENT"
    CLOCK_DOMAINS => "INDEPENDENT",

```

```
DOB_REG => 1,
-- Collision check: "ALL", "GENERATE_X_ONLY", "NONE", "WARNING_ONLY"
SIM_COLLISION_CHECK => "ALL",
-- DOA_REG, DOB_REG: Optional output register (0, 1)
DOA_REG => 1,
-- ENADDRENA/ENADDRENB: Address enable pin enable, "TRUE", "FALSE"
ENADDRENA => "FALSE",
ENADDRENB => "FALSE",
-- EN_ECC_PIPE: ECC pipeline register, "TRUE"/"FALSE"
EN_ECC_PIPE => "FALSE",
-- EN_ECC_READ: Enable ECC decoder, "TRUE"/"FALSE"
EN_ECC_READ => "FALSE",
-- EN_ECC_WRITE: Enable ECC encoder, "TRUE"/"FALSE"
EN_ECC_WRITE => "FALSE",
-- INITP_00 to INITP_0F: 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",
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 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_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"0000000000",
INIT_B => X"0000000000",
-- Initialization File: RAM initialization file
INIT_FILE => "NONE",
-- Programmable Inversion Attributes: Specifies the use of the built-in programmable inversion
IS_CLKARDCLK_INVERTED => '0',
IS_CLKBWRCLK_INVERTED => '0',
IS_ENARDEN_INVERTED => '0',
IS_ENBWREN_INVERTED => '0',
IS_RSTRAMARSTRAM_INVERTED => '0',
IS_RSTRAMB_INVERTED => '0',
IS_RSTREGARSTREG_INVERTED => '0',
IS_RSTREGB_INVERTED => '0',
-- RDADDRCHANGE: Disable memory access when output value does not change ("TRUE", "FALSE")
RDADDRCHANGEA => "FALSE",
RDADDRCHANGEB => "FALSE",
-- READ_WIDTH A/B, WRITE_WIDTH A/B: Read/write width per port
READ_WIDTH_A => 0, -- 0-9
READ_WIDTH_B => 0, -- 0-9
WRITE_WIDTH_A => 0, -- 0-9
WRITE_WIDTH_B => 0, -- 0-9
-- RSTREG_PRIORITY_A, RSTREG_PRIORITY_B: Reset or enable priority ("RSTREG", "REGCE")
RSTREG_PRIORITY_A => "RSTREG",
RSTREG_PRIORITY_B => "RSTREG",
-- SRVAL A, SRVAL B: Set/reset value for output
SRVAL_A => X"0000000000",
SRVAL_B => X"0000000000",
-- Sleep Async: Sleep function asynchronous or synchronous ("TRUE", "FALSE")
SLEEP_ASYNC => "FALSE",
-- WriteMode: "WRITE_FIRST", "NO_CHANGE", "READ_FIRST"
WRITE_MODE_A => "NO_CHANGE",
WRITE_MODE_B => "NO_CHANGE"
)
port map (
-- Cascade Signals outputs: Multi-BRAM cascade signals
CASDOUTA => CASDOUTA, -- 32-bit output: Port A cascade output data
CASDOUTB => CASDOUTB, -- 32-bit output: Port B cascade output data
CASDOUTPA => CASDOUTPA, -- 4-bit output: Port A cascade output parity data
CASDOUTPB => CASDOUTPB, -- 4-bit output: Port B cascade output parity data
CASOUTDBITERR => CASOUTDBITERR, -- 1-bit output: DBITERR cascade output
CASOUTSBITERR => CASOUTSBITERR, -- 1-bit output: SBITERR cascade output
-- ECC Signals outputs: Error Correction Circuitry ports
DBITERR => DBITERR, -- 1-bit output: Double bit error status
ECCPARITY => ECCPARITY, -- 8-bit output: Generated error correction parity
RDADRECC => RDADRECC, -- 9-bit output: ECC Read Address
SBITERR => SBITERR, -- 1-bit output: Single bit error status
-- Port A Data outputs: Port A data
DOUTADOUT => DOUTADOUT, -- 32-bit output: Port A ata/LSB data
DOUTPADOUTP => DOUTPADOUTP, -- 4-bit output: Port A parity/LSB parity
-- Port B Data outputs: Port B data
DOUTBDOUT => DOUTBDOUT, -- 32-bit output: Port B data/MSB data
DOUTPBDOUTP => DOUTPBDOUTP, -- 4-bit output: Port B parity/MSB parity
-- Cascade Signals inputs: Multi-BRAM cascade signals
CASDIMUXA => CASDIMUXA, -- 1-bit input: Port A input data (0=DINA, 1=CASDINA)
CASDIMUXB => CASDIMUXB, -- 1-bit input: Port B input data (0=DINB, 1=CASDINB)

```

```

CASDINA => CASDINA, -- 32-bit input: Port A cascade input data
CASDINB => CASDINB, -- 32-bit input: Port B cascade input data
CASDINPA => CASDINPA, -- 4-bit input: Port A cascade input parity data
CASDINPB => CASDINPB, -- 4-bit input: Port B cascade input parity data
CASDOMUXA => CASDOMUXA, -- 1-bit input: Port A unregistered data (0=BRAM data, 1=CASDINA)
CASDOMUXB => CASDOMUXB, -- 1-bit input: Port B unregistered data (0=BRAM data, 1=CASDINB)
CASDOMUXEN_A => CASDOMUXEN_A, -- 1-bit input: Port A unregistered output data enable
CASDOMUXEN_B => CASDOMUXEN_B, -- 1-bit input: Port B unregistered output data enable
CASINDBITERR => CASINDBITERR, -- 1-bit input: DBITERR cascade input
CASINSBITERR => CASINSBITERR, -- 1-bit input: SBITERR cascade input
CASOREGIMUXA => CASOREGIMUXA, -- 1-bit input: Port A registered data (0=BRAM data, 1=CASDINA)
CASOREGIMUXB => CASOREGIMUXB, -- 1-bit input: Port B registered data (0=BRAM data, 1=CASDINB)
CASOREGIMUXEN_A => CASOREGIMUXEN_A, -- 1-bit input: Port A registered output data enable
CASOREGIMUXEN_B => CASOREGIMUXEN_B, -- 1-bit input: Port B registered output data enable
-- ECC Signals inputs: Error Correction Circuitry ports
ECCPIPECE => ECCPIPECE, -- 1-bit input: ECC Pipeline Register Enable
INJECTDBITERR => INJECTDBITERR, -- 1-bit input: Inject a double bit error
INJECTSBITERR => INJECTSBITERR,
-- Port A Address/Control Signals inputs: Port A address and control signals
ADDRARDADDR => ADDRARDADDR, -- 15-bit input: A/Read port address
ADDRENA => ADDRENA, -- 1-bit input: Active-High A/Read port address enable
CLKARDCLK => CLKARDCLK, -- 1-bit input: A/Read port clock
ENARDEN => ENARDEN, -- 1-bit input: Port A enable/Read enable
REGCEAREGCE => REGCEAREGCE, -- 1-bit input: Port A register enable/Register enable
RSTRAMARSTRAM => RSTRAMARSTRAM, -- 1-bit input: Port A set/reset
RSTREGARSTREG => RSTREGARSTREG, -- 1-bit input: Port A register set/reset
SLEEP => SLEEP, -- 1-bit input: Sleep Mode
WEA => WEA, -- 4-bit input: Port A write enable
-- Port A Data inputs: Port A data
DINADIN => DINADIN, -- 32-bit input: Port A data/LSB data
DINPADINP => DINPADINP, -- 4-bit input: Port A parity/LSB parity
-- Port B Address/Control Signals inputs: Port B address and control signals
ADDRBWRADDR => ADDRBWRADDR, -- 15-bit input: B/Write port address
ADDRBENB => ADDRBENB, -- 1-bit input: Active-High B/Write port address enable
CLKBWRCLK => CLKBWRCLK, -- 1-bit input: B/Write port clock
ENBWREN => ENBWREN, -- 1-bit input: Port B enable/Write enable
REGCEB => REGCEB, -- 1-bit input: Port B register enable
RSTRAMB => RSTRAMB, -- 1-bit input: Port B set/reset
RSTREGB => RSTREGB, -- 1-bit input: Port B register set/reset
WEBWE => WEBWE, -- 8-bit input: Port B write enable/Write enable
-- Port B Data inputs: Port B data
DINBDIN => DINBDIN, -- 32-bit input: Port B data/MSB data
DINPBDINP => DINPBDINP, -- 4-bit input: Port B parity/MSB parity
);

-- End of RAMB36E2_inst instantiation
    
```

## Verilog Instantiation Template

```

// RAMB36E2: 36K-bit Configurable Synchronous Block RAM
// UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

RAMB36E2 #(
    // CASCADE_ORDER_A, CASCADE_ORDER_B: "FIRST", "MIDDLE", "LAST", "NONE"
    .CASCADE_ORDER_A("NONE"),
    .CASCADE_ORDER_B("NONE"),
    // CLOCK_DOMAINS: "COMMON", "INDEPENDENT"
    .CLOCK_DOMAINS("INDEPENDENT"),
    .DOB_REG(1),
    // Collision check: "ALL", "GENERATE_X_ONLY", "NONE", "WARNING_ONLY"
    .SIM_COLLISION_CHECK("ALL"),
    // DOA_REG, DOB_REG: Optional output register (0, 1)
    .DOA_REG(1),
    // ENADDRENA/ENADDRENB: Address enable pin enable, "TRUE", "FALSE"
    .ENADDRENA("FALSE"),
    .ENADDRENB("FALSE"),
    // EN_ECC_PIPE: ECC pipeline register, "TRUE"/"FALSE"
    .EN_ECC_PIPE("FALSE"),
    // EN_ECC_READ: Enable ECC decoder, "TRUE"/"FALSE"
    .EN_ECC_READ("FALSE")
)
    
```









```

.CASOREGIMUXB(CASOREGIMUXB),          // 1-bit input: Port B registered data (0=BRAM data, 1=CASDINB)
.CASOREGIMUXEN_A(CASOREGIMUXEN_A),    // 1-bit input: Port A registered output data enable
.CASOREGIMUXEN_B(CASOREGIMUXEN_B),    // 1-bit input: Port B registered output data enable
// ECC Signals inputs: Error Correction Circuitry ports
.ECCPIPECE(ECCPIPECE),                // 1-bit input: ECC Pipeline Register Enable
.INJECTDBITERR(INJECTDBITERR),        // 1-bit input: Inject a double bit error
.INJECTSBITERR(INJECTSBITERR),
// Port A Address/Control Signals inputs: Port A address and control signals
.ADDRDADDR(ADDRDADDR),                // 15-bit input: A/Read port address
.ADDRENA(ADDRENA),                    // 1-bit input: Active-High A/Read port address enable
.CLKARDCLK(CLKARDCLK),                 // 1-bit input: A/Read port clock
.ENARDEN(ENARDEN),                     // 1-bit input: Port A enable/Read enable
.REGCEAREGCE(REGCEAREGCE),            // 1-bit input: Port A register enable/Register enable
.RSTRAMARSTRAM(RSTRAMARSTRAM),        // 1-bit input: Port A set/reset
.RSTREGARSTREG(RSTREGARSTREG),        // 1-bit input: Port A register set/reset
.SLEEP(SLEEP),                         // 1-bit input: Sleep Mode
.WEA(WEA),                              // 4-bit input: Port A write enable
// Port A Data inputs: Port A data
.DINADIN(DINADIN),                    // 32-bit input: Port A data/LSB data
.DINPADINP(DINPADINP),                 // 4-bit input: Port A parity/LSB parity
// Port B Address/Control Signals inputs: Port B address and control signals
.ADDRBRADDR(ADDRBRADDR),              // 15-bit input: B/Write port address
.ADDRENB(ADDRENB),                     // 1-bit input: Active-High B/Write port address enable
.CLKBWRCLK(CLKBWRCLK),                 // 1-bit input: B/Write port clock
.ENBWREN(ENBWREN),                     // 1-bit input: Port B enable/Write enable
.REGCEB(REGCEB),                       // 1-bit input: Port B register enable
.RSTRAMB(RSTRAMB),                     // 1-bit input: Port B set/reset
.RSTREGB(RSTREGB),                     // 1-bit input: Port B register set/reset
.WEBWE(WEBWE),                         // 8-bit input: Port B write enable/Write enable
// Port B Data inputs: Port B data
.DINBDIN(DINBDIN),                    // 32-bit input: Port B data/MSB data
.DINPBDINP(DINPBDINP),                 // 4-bit input: Port B parity/MSB parity
);

// End of RAMB36E2_inst instantiation
    
```

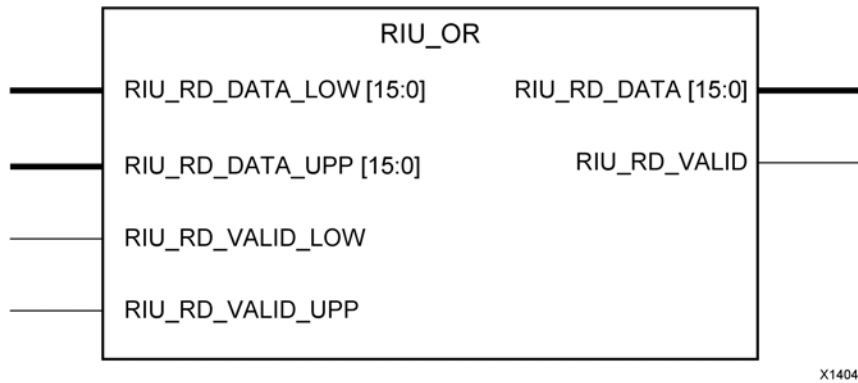
## For More Information

- See the *UltraScale Architecture Memory Resources User Guide* ([UG573](#)).
- See the [UltraScale Programmable Devices Documentation](#).

# RIU\_OR

## Primitive: Register Interface Unit Selection Block

PRIMITIVE\_GROUP: I/O  
 PRIMITIVE\_SUBGROUP: BITSlice  
 Families: UltraScale, UltraScale+



## Introduction

The RIU\_OR controls the Register Interface Unit selection between the lower and upper nibble BITSlice\_CONTROL within a BYTE.

## Port Descriptions

Port	Direction	Width	Function
RIU_RD_DATA<15:0>	Output	16	RIU data bus to the controller
RIU_RD_DATA_LOW<15:0>	Input	16	RIU data bus from the controller to the lower nibble BITSlice_CONTROL
RIU_RD_DATA_UPP<15:0>	Input	16	RIU data bus from the controller to the upper nibble BITSlice_CONTROL
RIU_RD_VALID	Output	1	Combined RIU read valid signal to the controller
RIU_RD_VALID_LOW	Input	1	RIU_VALID of the lower nibble BITSlice_CONTROL
RIU_RD_VALID_UPP	Input	1	RIU_VALID of the upper nibble BITSlice_CONTROL

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	Yes

## VHDL Instantiation Template

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RIU_OR: Register Interface Unit Selection Block
--      UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

RIU_OR_inst : RIU_OR
generic map (
    SIM_DEVICE => "ULTRASCALE" -- Set the device version (ULTRASCALE, ULTRASCALE_PLUS_ES1)
)
port map (
    RIU_RD_DATA => RIU_RD_DATA,          -- 16-bit output: RIU data bus to the controller
    RIU_RD_VALID => RIU_RD_VALID,        -- 1-bit output: Combined RIU read valid signal to the controller
    RIU_RD_DATA_LOW => RIU_RD_DATA_LOW,  -- 16-bit input: RIU data bus from the controller to the lower
                                         -- nibble BITSlice_CONTROL

    RIU_RD_DATA_UPP => RIU_RD_DATA_UPP,  -- 16-bit input: RIU data bus from the controller to the upper
                                         -- nibble BITSlice_CONTROL

    RIU_RD_VALID_LOW => RIU_RD_VALID_LOW, -- 1-bit input: RIU_VALID of the lower nibble BITSlice_CONTROL
    RIU_RD_VALID_UPP => RIU_RD_VALID_UPP -- 1-bit input: RIU_VALID of the upper nibble BITSlice_CONTROL
);

-- End of RIU_OR_inst instantiation
    
```

## Verilog Instantiation Template

```

// RIU_OR: Register Interface Unit Selection Block
//      UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

RIU_OR #(
    .SIM_DEVICE("ULTRASCALE") // Set the device version (ULTRASCALE, ULTRASCALE_PLUS_ES1)
)
RIU_OR_inst (
    .RIU_RD_DATA(RIU_RD_DATA),          // 16-bit output: RIU data bus to the controller
    .RIU_RD_VALID(RIU_RD_VALID),        // 1-bit output: Combined RIU read valid signal to the controller
    .RIU_RD_DATA_LOW(RIU_RD_DATA_LOW),  // 16-bit input: RIU data bus from the controller to the lower
                                         // nibble BITSlice_CONTROL

    .RIU_RD_DATA_UPP(RIU_RD_DATA_UPP),  // 16-bit input: RIU data bus from the controller to the upper
                                         // nibble BITSlice_CONTROL

    .RIU_RD_VALID_LOW(RIU_RD_VALID_LOW), // 1-bit input: RIU_VALID of the lower nibble BITSlice_CONTROL
    .RIU_RD_VALID_UPP(RIU_RD_VALID_UPP) // 1-bit input: RIU_VALID of the upper nibble BITSlice_CONTROL
);

// End of RIU_OR_inst instantiation
    
```

## For More Information

- See the *UltraScale Architecture SelectIO Resources User Guide* ([UG571](#)).
- See the [UltraScale Programmable Devices Documentation](#).

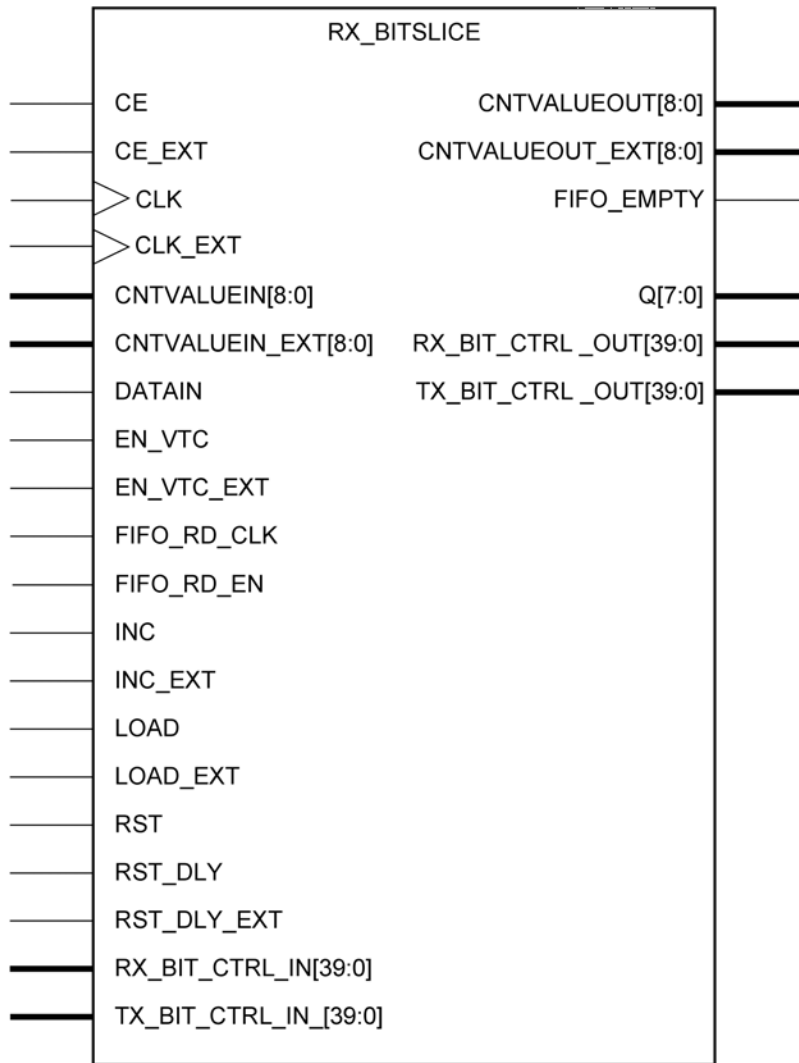
# RX\_BITSLICE

Primitive: RX\_BITSLICE for input using Native Mode

PRIMITIVE\_GROUP: I/O

PRIMITIVE\_SUBGROUP: BITSlice

Families: UltraScale, UltraScale+



## Introduction

In native mode, the RX\_BITSLICE contains deserialization logic and 512-tap input delay (IDELAY) that can be continuously adjusted for VT variation. The RX\_BITSLICE contains deserialization logic for 1:4 or 1:8 deserialization and a shallow FIFO to allow connection to another clock domain.

## Port Descriptions

Port	Direction	Width	Function
CE	Input	1	Clock enable for RX_BITSLICE IDELAY register clock
CE_EXT	Input	1	Optional extended (cascaded delay) clock enable for IDELAY register clock
CLK	Input	1	Clock used to sample LOAD, CE, INC
CLK_EXT	Input	1	Optional extended (cascaded delay) delay clock used to sample LOAD, CE, and INC
CNTVALUEIN <8:0>	Input	9	Counter value from internal device logic for tap value to be loaded dynamically
CNTVALUEIN_EXT <8:0>	Input	9	Optional extended (cascaded delay) counter value from internal device logic for tap value to be loaded dynamically
CNTVALUEOUT <8:0>	Output	9	Counter value to going the internal device logic for monitoring tap value
CNTVALUEOUT_EXT <8:0>	Output	9	Optional extended (cascaded delay) counter value going to the internal device logic for monitoring tap value
DATAIN	Input	1	Input signal from IBUF
EN_VTC	Input	1	Enables IDELAYCTRL to keep stable delay over VT when set to HIGH. VT compensation disabled when set to LOW.
EN_VTC_EXT	Input	1	Enables IDELAYCTRL to keep stable delay over VT of the cascaded delay when set to HIGH. VT compensation disabled when set to LOW.
FIFO_EMPTY	Output	1	FIFO empty flag for the FIFO of this bit
FIFO_RD_CLK	Input	1	FIFO read clock for the FIFO of this bit
FIFO_RD_EN	Input	1	FIFO read enable for the FIFO of this bit
INC	Input	1	Increment the current delay tap setting
INC_EXT	Input	1	Optional extended (cascaded delay) increments the current delay tap setting
LOAD	Input	1	Load the CNTVALUEIN tap setting
LOAD_EXT	Input	1	Optional extended (cascaded delay) load the CNTVALUEIN_EXT tap setting
Q <7:0>	Output	8	Registered output data from FIFO
RST	Input	1	Asynchronous assert, synchronous deassert for RX_BITSLICE ISERDES
RST_DLY	Input	1	Reset the internal DELAY value to DELAY_VALUE
RST_DLY_EXT	Input	1	Optional extended (cascaded delay) reset delay to DELAY_VALUE_EXT
RX_BIT_CTRL_IN <39:0>	Input	40	Input bus from BITSlice_CONTROL
RX_BIT_CTRL_OUT <39:0>	Output	40	Output bus to BITSlice_CONTROL
TX_BIT_CTRL_IN <39:0>	Input	40	Input bus from BITSlice_CONTROL
TX_BIT_CTRL_OUT <39:0>	Output	40	Output bus to BITSlice_CONTROL

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	Yes

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
CASCADE	STRING	"FALSE", "TRUE"	"FALSE"	<ul style="list-style-type: none"> <li>"TRUE": Enables cascading of IDELAY and ODELAY lines to get a total of 2.5ns delay. Extended delay controlled by the _EXT pins.</li> <li>"FALSE": Only use IDELAY delay line with a maximum of 1.25ns delay.</li> </ul>
DATA_TYPE	STRING	"DATA", "CLOCK", "DATA_AND_CLOCK", "SERIAL"	"DATA"	Defines whether the input pin is carrying a clock signal, a data signal, or clock signal that is also used as data.
DATA_WIDTH	DECIMAL	8, 4	8	Defines the width of the serial-to-parallel converter. Legal data widths are 4 and 8.
DELAY_FORMAT	STRING	"TIME", "COUNT"	"TIME"	Sets the units of DELAY_VALUE of the RX_BITSLICE IDELAY. Use TIME when DELAY_TYPE is FIXED. Use COUNT when DELAY_TYPE is VARIABLE or VAR_LOAD. <ul style="list-style-type: none"> <li>"TIME" : RX_BITSLICE IDELAY DELAY_VALUE is specified in ps</li> <li>"COUNT" : RX_BITSLICE IDELAY DELAY_VALUE is specified in taps</li> </ul>
DELAY_TYPE	STRING	"FIXED", "VAR_LOAD", "VARIABLE"	"FIXED"	Sets the type of tap delay line. <ul style="list-style-type: none"> <li>"FIXED" - Sets a static delay value</li> <li>"VARIABLE" - Dynamically adjust (increment/decrement) delay value</li> <li>"VAR_LOAD" - Dynamically loads tap values</li> </ul>
DELAY_VALUE	DECIMAL	0 to 1250	0	Specifies the fixed delay in ps when using FIXED DELAY_TYPE. Specifies value upon reset when using VARIABLE or VAR_LOAD.
DELAY_VALUE_EXT	DECIMAL	0 to 1250	0	Value of the extended input delay value in ps

Attribute	Type	Allowed Values	Default	Description
FIFO_SYNC_MODE	STRING	"FALSE", "TRUE"	"FALSE"	Internal write clock and FIFO_RD_CLK are coming from the same source <ul style="list-style-type: none"> <li>"TRUE" - Internal write clock and the FIFO_RD_CLK are coming from a common source.</li> <li>"FALSE" - FIFO write clock and the FIFO read clock are not on a common clock domain</li> </ul>
IS_CLK_EXT_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether the clock CLK_EXT pin is active-high or active-low.
IS_CLK_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether the clock CLK pin is active-high or active-low.
IS_RST_DLY_EXT_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether the reset RST_DLY_EXT pin is active-high or active-low.
IS_RST_DLY_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether the reset RST_DLY pin is active-high or active-low.
IS_RST_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether the reset RST pin is active-high or active-low.
REFCLK_FREQUENCY	1 significant digit FLOAT	200.0 to 2667.0	300.0	Specification of the reference clock frequency in MHz
SIM_DEVICE	STRING	"ULTRASCALE", "ULTRASCALE_PLUS_ES1"	"ULTRASCALE"	Set the device version
UPDATE_MODE	STRING	"ASYNC", "MANUAL", "SYNC"	"ASYNC"	<ul style="list-style-type: none"> <li>"ASYNC": Updates are increments or decrements to the delay value independent of the data being received</li> <li>"SYNC": Updates require that data input transitions to synchronously update the delay with the data input edges</li> <li>"MANUAL": Updates take effect when both LOAD and CE are asserted after the LOAD and CNTVALUEIN signals are used to load the new CNTVALUE</li> </ul>



Attribute	Type	Allowed Values	Default	Description
UPDATE_MODE_EXT	STRING	"ASYNC", "MANUAL", "SYNC"	"ASYNC"	<ul style="list-style-type: none"> <li>"ASYNC": Updates are increments or decrements to the delay value independent of the data being received</li> <li>"SYNC": Updates require that data input transitions to synchronously update the delay with the data input edges</li> <li>"MANUAL": Updates take effect when both LOAD and CE are asserted after the LOAD and CNTVALUEIN signals are used to load the new CNTVALUE</li> </ul>

## VHDL Instantiation Template

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RX_BITSLICE: RX_BITSLICE for input using Native Mode
-- UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

RX_BITSLICE_inst : RX_BITSLICE
generic map (
    CASCADE => "FALSE",           -- Enables cascading of IDELAY and ODELAY lines
    DATA_TYPE => "DATA",         -- Defines what the input pin is carrying (CLOCK, DATA, DATA_AND_CLOCK,
    -- SERIAL)
    DATA_WIDTH => 8,             -- Defines the width of the serial-to-parallel converter (4-8)
    DELAY_FORMAT => "TIME",       -- Units of the DELAY_VALUE (COUNT, TIME)
    DELAY_TYPE => "FIXED",        -- Set the type of tap delay line (FIXED, VARIABLE, VAR_LOAD)
    DELAY_VALUE => 0,             -- Input delay value setting in ps
    DELAY_VALUE_EXT => 0,         -- Value of the extended input delay value in ps
    FIFO_SYNC_MODE => "FALSE",    -- Internal write clock and FIFO_RD_CLK are coming from a common source
    IS_CLK_EXT_INVERTED => '0',   -- Optional inversion for CLK_EXT
    IS_CLK_INVERTED => '0',       -- Optional inversion for CLK
    IS_RST_DLY_EXT_INVERTED => '0', -- Optional inversion for RST_DLY_EXT
    IS_RST_DLY_INVERTED => '0',   -- Optional inversion for RST_DLY
    IS_RST_INVERTED => '0',       -- Optional inversion for RST
    REFCLK_FREQUENCY => 300.0,    -- Specification of the reference clock frequency in MHz (200.0-2667.0)
    SIM_DEVICE => "ULTRASCALE",   -- Set the device version (ULTRASCALE, ULTRASCALE_PLUS_ES1)
    UPDATE_MODE => "ASYNC",       -- Determines when updates to the delay will take effect (ASYNC, MANUAL,
    -- SYNC)
    UPDATE_MODE_EXT => "ASYNC"    -- Determines when updates to the extended input delay will take effect
    -- (ASYNC, MANUAL, SYNC)
)
port map (
    CNTVALUEOUT => CNTVALUEOUT,   -- 9-bit output: Counter value to device logic
    CNTVALUEOUT_EXT => CNTVALUEOUT_EXT, -- 9-bit output: Optional extended (cascaded delay) counter value
    -- going to the device logic

    FIFO_EMPTY => FIFO_EMPTY,    -- 1-bit output: FIFO empty flag
    Q => Q,                        -- 8-bit output: Registered output data from FIFO
    RX_BIT_CTRL_OUT => RX_BIT_CTRL_OUT, -- 40-bit output: Output bus to BITSlice_CONTROL
    TX_BIT_CTRL_OUT => TX_BIT_CTRL_OUT, -- 40-bit output: Output bus to BITSlice_CONTROL
    CE => CE,                      -- 1-bit input: Clock enable for IDELAY
    CE_EXT => CE_EXT,              -- 1-bit input: Optional extended (cascaded delay) clock enable

```

```

CLK => CLK,                -- 1-bit input: Clock used to sample LOAD, CE, INC
CLK_EXT => CLK_EXT,        -- 1-bit input: Optional extended (cascaded delay) clock
CNTVALUEIN => CNTVALUEIN, -- 9-bit input: Counter value from device logic
CNTVALUEIN_EXT => CNTVALUEIN_EXT, -- 9-bit input: Optional extended (cascaded delay) counter value from
                                -- device logic

DATAIN => DATAIN,        -- 1-bit input: Input signal from IBUF
EN_VTC => EN_VTC,        -- 1-bit input: Enable IDELAYCTRL to keep stable delay over VT
EN_VTC_EXT => EN_VTC_EXT, -- 1-bit input: Optional extended (cascaded delay) to keep stable
                                -- delay over VT

FIFO_RD_CLK => FIFO_RD_CLK, -- 1-bit input: FIFO read clock
FIFO_RD_EN => FIFO_RD_EN,  -- 1-bit input: FIFO read enable
INC => INC,                -- 1-bit input: Increment the current delay tap setting
INC_EXT => INC_EXT,        -- 1-bit input: Optional extended (cascaded delay) increments the
                                -- current delay tap setting

LOAD => LOAD,             -- 1-bit input: Load the CNTVALUEIN tap setting
LOAD_EXT => LOAD_EXT,     -- 1-bit input: Optional extended (cascaded delay) load the
                                -- CNTVALUEIN_EXT tap setting

RST => RST,               -- 1-bit input: Asynchronous assert, synchronous deassert for
                                -- RX_BITSLICE ISERDES

RST_DLY => RST_DLY,       -- 1-bit input: Reset the internal DELAY value to DELAY_VALUE
RST_DLY_EXT => RST_DLY_EXT, -- 1-bit input: Optional extended (cascaded delay) reset delay to
                                -- DELAY_VALUE_EXT

RX_BIT_CTRL_IN => RX_BIT_CTRL_IN, -- 40-bit input: Input bus from BITSlice_CONTROL
TX_BIT_CTRL_IN => TX_BIT_CTRL_IN  -- 40-bit input: Input bus from BITSlice_CONTROL
);

-- End of RX_BITSLICE_inst instantiation
    
```

## Verilog Instantiation Template

```

// RX_BITSLICE: RX_BITSLICE for input using Native Mode
//                               UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

RX_BITSLICE #(
    .CASCADE("FALSE"),          // Enables cascading of IDELAY and ODELAY lines
    .DATA_TYPE("DATA"),        // Defines what the input pin is carrying (CLOCK, DATA, DATA_AND_CLOCK,
                                // SERIAL)
    .DATA_WIDTH(8),            // Defines the width of the serial-to-parallel converter (4-8)
    .DELAY_FORMAT("TIME"),     // Units of the DELAY_VALUE (COUNT, TIME)
    .DELAY_TYPE("FIXED"),      // Set the type of tap delay line (FIXED, VARIABLE, VAR_LOAD)
    .DELAY_VALUE(0),           // Input delay value setting in ps
    .DELAY_VALUE_EXT(0),       // Value of the extended input delay value in ps
    .FIFO_SYNC_MODE("FALSE"),  // Internal write clock and FIFO_RD_CLK are coming from a common source
    .IS_CLK_EXT_INVERTED(1'b0), // Optional inversion for CLK_EXT
    .IS_CLK_INVERTED(1'b0),    // Optional inversion for CLK
    .IS_RST_DLY_EXT_INVERTED(1'b0), // Optional inversion for RST_DLY_EXT
    .IS_RST_DLY_INVERTED(1'b0), // Optional inversion for RST_DLY
    .IS_RST_INVERTED(1'b0),    // Optional inversion for RST
    .REFCLK_FREQUENCY(300.0),  // Specification of the reference clock frequency in MHz (200.0-2667.0)
    .SIM_DEVICE("ULTRASCALE"), // Set the device version (ULTRASCALE, ULTRASCALE_PLUS_ES1)
    .UPDATE_MODE("ASYNCR"),    // Determines when updates to the delay will take effect (ASYNCR, MANUAL,
                                // SYNC)
    .UPDATE_MODE_EXT("ASYNCR") // Determines when updates to the extended input delay will take effect
                                // (ASYNCR, MANUAL, SYNC)
)
RX_BITSLICE_inst (
    .CNTVALUEOUT(CNTVALUEOUT), // 9-bit output: Counter value to device logic
    .CNTVALUEOUT_EXT(CNTVALUEOUT_EXT), // 9-bit output: Optional extended (cascaded delay) counter value
                                        // going to the device logic

    .FIFO_EMPTY(FIFO_EMPTY), // 1-bit output: FIFO empty flag
    .Q(Q),                    // 8-bit output: Registered output data from FIFO
    .RX_BIT_CTRL_OUT(RX_BIT_CTRL_OUT), // 40-bit output: Output bus to BITSlice_CONTROL
    
```

```

.TX_BIT_CTRL_OUT(TX_BIT_CTRL_OUT), // 40-bit output: Output bus to BITSlice_CONTROL
.CE(CE), // 1-bit input: Clock enable for IDELAY
.CE_EXT(CE_EXT), // 1-bit input: Optional extended (cascaded delay) clock enable
.CLK(CLK), // 1-bit input: Clock used to sample LOAD, CE, INC
.CLK_EXT(CLK_EXT), // 1-bit input: Optional extended (cascaded delay) clock
.CNTVALUEIN(CNTVALUEIN), // 9-bit input: Counter value from device logic
.CNTVALUEIN_EXT(CNTVALUEIN_EXT), // 9-bit input: Optional extended (cascaded delay) counter value from
// device logic

.DATAIN(DATAIN), // 1-bit input: Input signal from IBUF
.EN_VTC(EN_VTC), // 1-bit input: Enable IDELAYCTRL to keep stable delay over VT
.EN_VTC_EXT(EN_VTC_EXT), // 1-bit input: Optional extended (cascaded delay) to keep stable
// delay over VT

.FIFO_RD_CLK(FIFO_RD_CLK), // 1-bit input: FIFO read clock
.FIFO_RD_EN(FIFO_RD_EN), // 1-bit input: FIFO read enable
.INC(INC), // 1-bit input: Increment the current delay tap setting
.INC_EXT(INC_EXT), // 1-bit input: Optional extended (cascaded delay) increments the
// current delay tap setting

.LOAD(LOAD), // 1-bit input: Load the CNTVALUEIN tap setting
.LOAD_EXT(LOAD_EXT), // 1-bit input: Optional extended (cascaded delay) load the
// CNTVALUEIN_EXT tap setting

.RST(RST), // 1-bit input: Asynchronous assert, synchronous deassert for
// RX_BITSLICE ISERDES

.RST_DLY(RST_DLY), // 1-bit input: Reset the internal DELAY value to DELAY_VALUE
.RST_DLY_EXT(RST_DLY_EXT), // 1-bit input: Optional extended (cascaded delay) reset delay to
// DELAY_VALUE_EXT

.RX_BIT_CTRL_IN(RX_BIT_CTRL_IN), // 40-bit input: Input bus from BITSlice_CONTROL
.TX_BIT_CTRL_IN(TX_BIT_CTRL_IN) // 40-bit input: Input bus from BITSlice_CONTROL
);

// End of RX_BITSLICE_inst instantiation
    
```

## For More Information

- See the *UltraScale Architecture SelectIO Resources User Guide* ([UG571](#)).
- See the [UltraScale Programmable Devices Documentation](#).

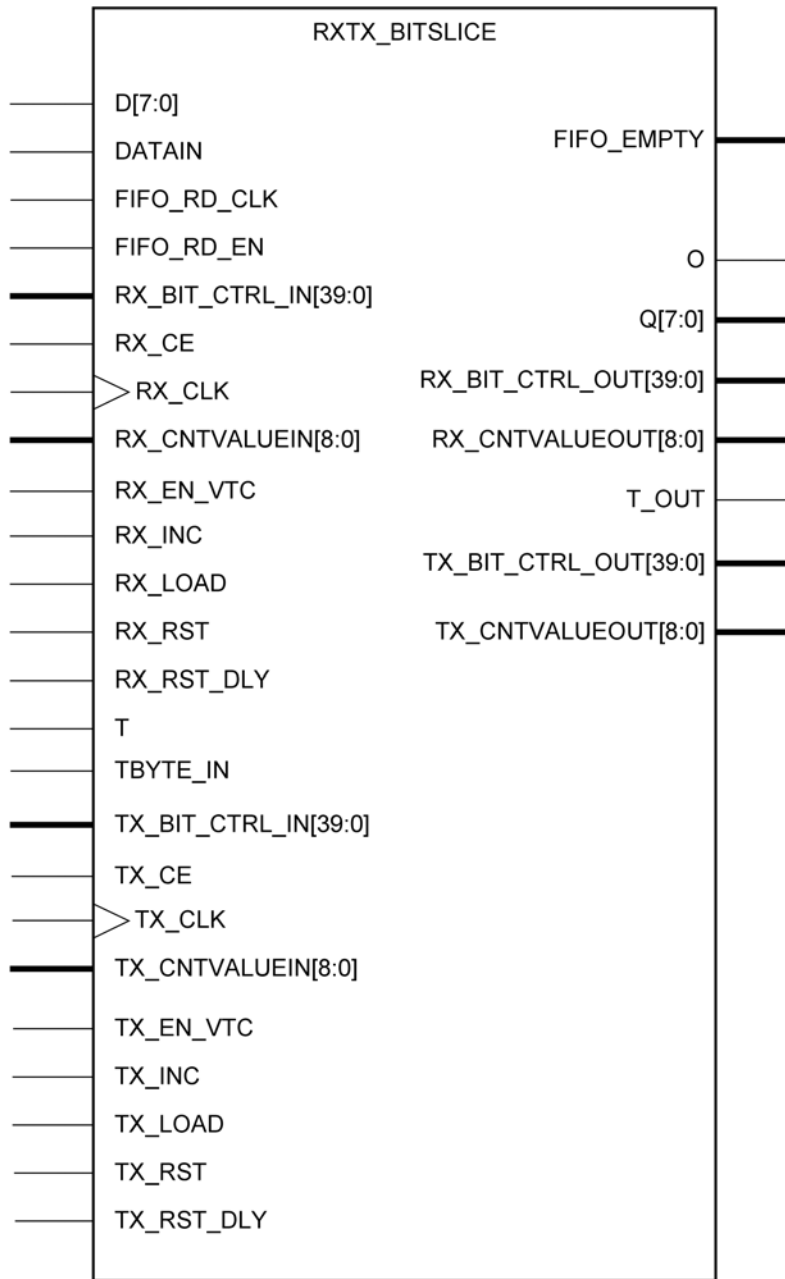
## RXTX\_BITSLICE

Primitive: RXTX\_BITSLICE for bidirectional I/O using Native Mode

PRIMITIVE\_GROUP: I/O

PRIMITIVE\_SUBGROUP: BITSlice

Families: UltraScale, UltraScale+



X14043

## Introduction

In native mode, the RXTX\_BITSLICE performs both functions of an RX\_BITSLICE and TX\_BITSLICE for bidirectional I/O.

## Port Descriptions

Port	Direction	Width	Function
D<7:0>	Input	8	Data from device logic
DATAIN	Input	1	Input signal from IOBUF
FIFO_EMPTY	Output	1	FIFO empty flag for the FIFO of this bit
FIFO_RD_CLK	Input	1	FIFO read clock for the FIFO of this bit
FIFO_RD_EN	Input	1	FIFO read enable for the FIFO of this bit
O	Output	1	Serialized output going to output buffer
Q<7:0>	Output	8	Registered output data from FIFO
RX_BIT_CTRL_IN<39:0>	Input	40	RX Input bus from BITSlice_CONTROL
RX_BIT_CTRL_OUT<39:0>	Output	40	RX Output bus to BITSlice_CONTROL
RX_CE	Input	1	Clock enable for RXTX_BITSLICE IDELAY register clock
RX_CLK	Input	1	RX Clock used to sample LOAD, CE, INC
RX_CNTVALUEIN<8:0>	Input	9	RX Counter value from internal device logic for tap value to be loaded dynamically
RX_CNTVALUEOUT<8:0>	Output	9	RX Counter value from internal device logic for tap value to be loaded dynamically
RX_EN_VTC	Input	1	RX Enable to keep stable delay over VT when set to HIGH. VT compensation disabled when set to LOW.
RX_INC	Input	1	RX Increment the current delay tap setting
RX_LOAD	Input	1	RX Load the CNTVALUEIN tap setting
RX_RST	Input	1	RX Asynchronous assert, synchronous deassert for RXTX_BITSLICE ISERDES
RX_RST_DLY	Input	1	RX Reset the internal DELAY value to DELAY_VALUE
T	Input	1	Legacy T byte input from device logic
T_OUT	Output	1	Byte group 3-state output
TBYTE_IN	Input	1	Byte group 3-state input from TX_BITSLICE_TRI
TX_BIT_CTRL_IN<39:0>	Input	40	TX Input bus from BITSlice_CONTROL
TX_BIT_CTRL_OUT<39:0>	Output	40	Output bus to BITSlice_CONTROL for TX
TX_CE	Input	1	Clock enable for RXTX_BITSLICE ODELAY register clock
TX_CLK	Input	1	TX Clock used to sample LOAD, CE, INC
TX_CNTVALUEIN<8:0>	Input	9	TX Counter value from internal device logic for tap value to be loaded dynamically
TX_CNTVALUEOUT<8:0>	Output	9	TX Counter value to going the internal device logic for monitoring tap value

Port	Direction	Width	Function
TX_EN_VTC	Input	1	TX Enable to keep stable delay over VT when set to HIGH. VT compensation disabled when set to LOW.
TX_INC	Input	1	TX Increment the current delay tap setting
TX_LOAD	Input	1	TX Load the CNTVALUEIN tap setting
TX_RST	Input	1	TX Asynchronous assert, synchronous deassert for RXTX_BITSLICE OSERDES
TX_RST_DLY	Input	1	TX Reset the internal DELAY value to DELAY_VALUE

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	Yes

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
ENABLE_PRE_EMPHASIS	STRING	"FALSE", "TRUE"	"FALSE"	Used in conjunction with IOB to Enable the pre-emphasis.
FIFO_SYNC_MODE	STRING	"FALSE", "TRUE"	"FALSE"	Internal write clock and FIFO_RD_CLK are coming from the same source <ul style="list-style-type: none"> <li>"TRUE" - Internal write clock and the FIFO_RD_CLK are coming from a common source.</li> <li>"FALSE" - FIFO write clock and the FIFO read clock are not on a common clock domain</li> </ul>
INIT	BINARY	1'b1, 1'b0	1'b1	Defines initial O value
IS_RX_CLK_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether the clock RX_CLK pin is active-high or active-low.
IS_RX_RST_DLY_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether the reset RX_RST_DLY pin is active-high or active-low.
IS_RX_RST_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether the reset RX_RST pin is active-high or active-low.
IS_TX_CLK_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether the clock TX_CLK pin is active-high or active-low.

Attribute	Type	Allowed Values	Default	Description
IS_TX_RST_DLY_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether the reset TX_RST_DLY pin is active-high or active-low.
IS_TX_RST_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether the reset TX_RST pin is active-high or active-low.
RX_DATA_TYPE	STRING	"DATA", "CLOCK", "DATA_AND_CLOCK", "SERIAL"	"DATA"	Defines whether the RX input pin is carrying a clock signal, a data signal, or clock signal that is also used as data.
RX_DATA_WIDTH	DECIMAL	8, 4	8	Defines the width of the serial-to-parallel converter. Legal data widths are 4 and 8.
RX_DELAY_FORMAT	STRING	"TIME", "COUNT"	"TIME"	Sets the units of DELAY_VALUE of the RXTX_BITSLICE IDELAY. Use TIME when DELAY_TYPE is FIXED. Use COUNT when DELAY_TYPE is VARIABLE or VAR_LOAD. <ul style="list-style-type: none"> <li>"TIME" : RXTX_BITSLICE IDELAY DELAY_VALUE is specified in ps</li> <li>"COUNT" : RXTX_BITSLICE IDELAY DELAY_VALUE is specified in taps</li> </ul>
RX_DELAY_TYPE	STRING	"FIXED", "VAR_LOAD", "VARIABLE"	"FIXED"	Sets the type of RX tap delay line. <ul style="list-style-type: none"> <li>"FIXED" - Sets a static delay value</li> <li>"VARIABLE" - Dynamically adjust (increment/decrement) delay value</li> <li>"VAR_LOAD" - Dynamically loads tap values</li> </ul>
RX_DELAY_VALUE	DECIMAL	0 to 1250	0	Specifies the RX fixed delay in ps when using FIXED DELAY_TYPE. Specifies RX value upon reset when using VARIABLE or VAR_LOAD.
RX_REFCLK_FREQUENCY	1 significant digit FLOAT	200.0 to 2667.0	300.0	Specification of the RX reference clock frequency in MHz

Attribute	Type	Allowed Values	Default	Description
RX_UPDATE_MODE	STRING	"ASYNC", "MANUAL", "SYNC"	"ASYNC"	<ul style="list-style-type: none"> <li>"ASYNC": Updates are increments or decrements to the delay value independent of the data being received</li> <li>"SYNC": Updates require that data input transitions to synchronously update the delay with the data input edges</li> <li>"MANUAL": Updates take effect when both LOAD and CE are asserted after the LOAD and CNTVALUEIN signals are used to load the new CNTVALUE</li> </ul>
SIM_DEVICE	STRING	"ULTRASCALE", "ULTRASCALE_PLUS_ES1"	"ULTRASCALE"	Set the device version
TBYTE_CTL	STRING	"TBYTE_IN", "T"	"TBYTE_IN"	Select between T and TBYTE_IN inputs in OSERDES mode only
TX_DATA_WIDTH	DECIMAL	8, 4	8	Defines the parallel data input width. Legal values are 4 and 8.
TX_DELAY_FORMAT	STRING	"TIME", "COUNT"	"TIME"	Sets the units of DELAY_VALUE of the RXTX_BITSLICE ODELAY. Use TIME when DELAY_TYPE is FIXED. Use COUNT when DELAY_TYPE is VARIABLE or VAR_LOAD. <ul style="list-style-type: none"> <li>"TIME" : RXTX_BITSLICE ODELAY DELAY_VALUE is specified in ps</li> <li>"COUNT" : RXTX_BITSLICE ODELAY DELAY_VALUE is specified in taps</li> </ul>
TX_DELAY_TYPE	STRING	"FIXED", "VAR_LOAD", "VARIABLE"	"FIXED"	Sets the type of TX tap delay line. <ul style="list-style-type: none"> <li>"FIXED" - Sets a static delay value</li> <li>"VARIABLE" - Dynamically adjust (increment/decrement) delay value</li> <li>"VAR_LOAD" - Dynamically loads tap values</li> </ul>
TX_DELAY_VALUE	DECIMAL	0 to 1250	0	Specifies the TX fixed delay in ps when using FIXED DELAY_TYPE. Specifies TX value upon reset when using VARIABLE or VAR_LOAD.
TX_OUTPUT_PHASE_90	STRING	"FALSE", "TRUE"	"FALSE"	Delays the output phase by 90-degrees



Attribute	Type	Allowed Values	Default	Description
TX_REFCLK_FREQUENCY	1 significant digit FLOAT	200.0 to 2667.0	300.0	Specification of the TX reference clock frequency in MHz
TX_UPDATE_MODE	STRING	"ASYNC", "MANUAL", "SYNC"	"ASYNC"	<ul style="list-style-type: none"> <li>"ASYNC": Updates are increments or decrements to the delay value independent of the data being received</li> <li>"SYNC": Updates require that input data transitions to synchronously update the delay with the data input edges</li> <li>"MANUAL": Updates take effect when both LOAD and CE are asserted after the LOAD and CNTVALUEIN signals are used to load the new CNTVALUE</li> </ul>

## VHDL Instantiation Template

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RXTX_BITSLICE: RXTX_BITSLICE for bidirectional I/O using Native Mode
--                               UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

RXTX_BITSLICE_inst : RXTX_BITSLICE
generic map (
    ENABLE_PRE_EMPHASIS => "FALSE", -- Enable the pre-emphasis
    FIFO_SYNC_MODE => "FALSE", -- Internal write clock and FIFO_RD_CLK are coming from a common source
    INIT => '1', -- Defines initial 0 value
    IS_RX_CLK_INVERTED => '0', -- Optional inversion for RX_CLK
    IS_RX_RST_DLY_INVERTED => '0', -- Optional inversion for RX_RST_DLY
    IS_RX_RST_INVERTED => '0', -- Optional inversion for RX_RST
    IS_TX_CLK_INVERTED => '0', -- Optional inversion for TX_CLK
    IS_TX_RST_DLY_INVERTED => '0', -- Optional inversion for TX_RST_DLY
    IS_TX_RST_INVERTED => '0', -- Optional inversion for TX_RST
    RX_DATA_TYPE => "DATA", -- Defines what the RX input pin is carrying (CLOCK, DATA,
    -- DATA AND CLOCK, SERIAL)
    RX_DATA_WIDTH => 8, -- Defines the width of the serial-to-parallel converter (4-8)
    RX_DELAY_FORMAT => "TIME", -- Units of the RX DELAY_VALUE (COUNT, TIME)
    RX_DELAY_TYPE => "FIXED", -- Set the type of RX tap delay line (FIXED, VARIABLE, VAR_LOAD)
    RX_DELAY_VALUE => 0, -- RX Input delay value setting in ps
    RX_REFCLK_FREQUENCY => 300.0, -- Specification of the RX reference clock frequency in MHz
    -- (200.0-2667.0)
    RX_UPDATE_MODE => "ASYNC", -- Determines when updates to the RX delay will take effect (ASYNC,
    -- MANUAL, SYNC)
    SIM_DEVICE => "ULTRASCALE", -- Set the device version (ULTRASCALE, ULTRASCALE_PLUS_ES1)
    TBYTE_CTL => "TBYTE_IN", -- Select between T and TBYTE_IN inputs
    TX_DATA_WIDTH => 8, -- Parallel data input width (4-8)
    TX_DELAY_FORMAT => "TIME", -- Units of the TX DELAY_VALUE (COUNT, TIME)
    TX_DELAY_TYPE => "FIXED", -- Set the type of TX tap delay line (FIXED, VARIABLE, VAR_LOAD)
    TX_DELAY_VALUE => 0, -- TX Input delay value setting in ps
    TX_OUTPUT_PHASE_90 => "FALSE", -- Delays the output phase by 90-degrees
    TX_REFCLK_FREQUENCY => 300.0, -- Specification of the TX reference clock frequency in MHz
    -- (200.0-2667.0)
    TX_UPDATE_MODE => "ASYNC" -- Determines when updates to the delay will take effect (ASYNC, MANUAL,

```

```

-- SYNC)
)
port map (
    FIFO_EMPTY => FIFO_EMPTY,           -- 1-bit output: FIFO empty flag
    O => O,                               -- 1-bit output: Serialized output going to output buffer
    Q => Q,                               -- 8-bit output: Registered output data from FIFO
    RX_BIT_CTRL_OUT => RX_BIT_CTRL_OUT,  -- 40-bit output: RX Output bus to BITSlice CONTROL
    RX_CNTVALUEOUT => RX_CNTVALUEOUT,    -- 9-bit output: RX Counter value from device logic
    TX_BIT_CTRL_OUT => TX_BIT_CTRL_OUT,  -- 40-bit output: Output bus to BITSlice CONTROL for TX
    TX_CNTVALUEOUT => TX_CNTVALUEOUT,    -- 9-bit output: TX Counter value to device logic
    T_OUT => T_OUT,                      -- 1-bit output: Byte group 3-state output
    D => D,                               -- 8-bit input: Data from device logic
    DATAIN => DATAIN,                  -- 1-bit input: Input signal from IOBUF
    FIFO_RD_CLK => FIFO_RD_CLK,          -- 1-bit input: FIFO read clock
    FIFO_RD_EN => FIFO_RD_EN,            -- 1-bit input: FIFO read enable
    RX_BIT_CTRL_IN => RX_BIT_CTRL_IN,    -- 40-bit input: RX Input bus from BITSlice CONTROL
    RX_CE => RX_CE,                      -- 1-bit input: Clock enable for IDELAY
    RX_CLK => RX_CLK,                    -- 1-bit input: RX Clock used to sample LOAD, CE, INC
    RX_CNTVALUEIN => RX_CNTVALUEIN,      -- 9-bit input: RX Counter value from device logic
    RX_EN_VTC => RX_EN_VTC,              -- 1-bit input: RX Enable to keep stable delay over VT
    RX_INC => RX_INC,                    -- 1-bit input: RX Increment the current delay tap setting
    RX_LOAD => RX_LOAD,                  -- 1-bit input: RX Load the CNTVALUEIN tap setting
    RX_RST => RX_RST,                    -- 1-bit input: RX Asynchronous assert, synchronous deassert for
                                         -- RXTX_BITSlice ISERDES

    RX_RST_DLY => RX_RST_DLY,            -- 1-bit input: RX Reset the internal DELAY value to DELAY_VALUE
    T => T,                               -- 1-bit input: Legacy T byte input from device logic
    TBYTE_IN => TBYTE_IN,                 -- 1-bit input: Byte group 3-state input from TX BITSlice TRI
    TX_BIT_CTRL_IN => TX_BIT_CTRL_IN,    -- 40-bit input: TX Input bus from BITSlice CONTROL
    TX_CE => TX_CE,                      -- 1-bit input: Clock enable for ODELAY
    TX_CLK => TX_CLK,                    -- 1-bit input: TX Clock used to sample LOAD, CE, INC
    TX_CNTVALUEIN => TX_CNTVALUEIN,      -- 9-bit input: TX Counter value from device logic
    TX_EN_VTC => TX_EN_VTC,              -- 1-bit input: TX Enable to keep stable delay over VT
    TX_INC => TX_INC,                    -- 1-bit input: TX Increment the current delay tap setting
    TX_LOAD => TX_LOAD,                  -- 1-bit input: TX Load the CNTVALUEIN tap setting
    TX_RST => TX_RST,                    -- 1-bit input: TX Asynchronous assert, synchronous deassert for
                                         -- RXTX_BITSlice OSERDES

    TX_RST_DLY => TX_RST_DLY             -- 1-bit input: TX Reset the internal DELAY value to DELAY_VALUE
);

-- End of RXTX_BITSlice_inst instantiation

```

## Verilog Instantiation Template

```

// RXTX_BITSlice: RXTX_BITSlice for bidirectional I/O using Native Mode
// UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

RXTX_BITSlice #(
    .ENABLE_PRE_EMPHASIS("FALSE"), // Enable the pre-emphasis
    .FIFO_SYNC_MODE("FALSE"),      // Internal write clock and FIFO_RD_CLK are coming from a common source
    .INIT(1'b1),                   // Defines initial 0 value
    .IS_RX_CLK_INVERTED(1'b0),     // Optional inversion for RX_CLK
    .IS_RX_RST_DLY_INVERTED(1'b0), // Optional inversion for RX_RST_DLY
    .IS_RX_RST_INVERTED(1'b0),     // Optional inversion for RX_RST
    .IS_TX_CLK_INVERTED(1'b0),     // Optional inversion for TX_CLK
    .IS_TX_RST_DLY_INVERTED(1'b0), // Optional inversion for TX_RST_DLY
    .IS_TX_RST_INVERTED(1'b0),     // Optional inversion for TX_RST
    .RX_DATA_TYPE("DATA"),         // Defines what the RX input pin is carrying (CLOCK, DATA, DATA_AND_CLOCK,
    // SERIAL)
    .RX_DATA_WIDTH(8),             // Defines the width of the serial-to-parallel converter (4-8)
    .RX_DELAY_FORMAT("TIME"),      // Units of the RX DELAY_VALUE (COUNT, TIME)
    .RX_DELAY_TYPE("FIXED"),       // Set the type of RX tap delay line (FIXED, VARIABLE, VAR_LOAD)
    .RX_DELAY_VALUE(0),            // RX Input delay value setting in ps
    .RX_REFCLK_FREQUENCY(300.0),   // Specification of the RX reference clock frequency in MHz (200.0-2667.0)
    .RX_UPDATE_MODE("ASYNC"),      // Determines when updates to the RX delay will take effect (ASYNC,
    // MANUAL, SYNC)
    .SIM_DEVICE("ULTRASCALE"),     // Set the device version (ULTRASCALE, ULTRASCALE_PLUS_ES1)
    .TBYTE_CTL("TBYTE_IN"),        // Select between T and TBYTE_IN inputs

```

```

        .TX_DATA_WIDTH(8), // Parallel data input width (4-8)
        .TX_DELAY_FORMAT("TIME"), // Units of the TX_DELAY_VALUE (COUNT, TIME)
        .TX_DELAY_TYPE("FIXED"), // Set the type of TX tap delay line (FIXED, VARIABLE, VAR_LOAD)
        .TX_DELAY_VALUE(0), // TX Input delay value setting in ps
        .TX_OUTPUT_PHASE_90("FALSE"), // Delays the output phase by 90-degrees
        .TX_REFCLK_FREQUENCY(300.0), // Specification of the TX reference clock frequency in MHz (200.0-2667.0)
        .TX_UPDATE_MODE("ASYNC") // Determines when updates to the delay will take effect (ASYN, MANUAL,
        // SYNC)
    )
    RXTX_BITSLICE_inst (
        .FIFO_EMPTY(FIFO_EMPTY), // 1-bit output: FIFO empty flag
        .O(O), // 1-bit output: Serialized output going to output buffer
        .Q(Q), // 8-bit output: Registered output data from FIFO
        .RX_BIT_CTRL_OUT(RX_BIT_CTRL_OUT), // 40-bit output: RX Output bus to BITSlice_CONTROL
        .RX_CNTVALUEOUT(RX_CNTVALUEOUT), // 9-bit output: RX Counter value from device logic
        .TX_BIT_CTRL_OUT(TX_BIT_CTRL_OUT), // 40-bit output: Output bus to BITSlice_CONTROL for TX
        .TX_CNTVALUEOUT(TX_CNTVALUEOUT), // 9-bit output: TX Counter value to device logic
        .T_OUT(T_OUT), // 1-bit output: Byte group 3-state output
        .D(D), // 8-bit input: Data from device logic
        .DATAIN(DATAIN), // 1-bit input: Input signal from IOBUF
        .FIFO_RD_CLK(FIFO_RD_CLK), // 1-bit input: FIFO read clock
        .FIFO_RD_EN(FIFO_RD_EN), // 1-bit input: FIFO read enable
        .RX_BIT_CTRL_IN(RX_BIT_CTRL_IN), // 40-bit input: RX Input bus from BITSlice_CONTROL
        .RX_CE(RX_CE), // 1-bit input: Clock enable for IDELAY
        .RX_CLK(RX_CLK), // 1-bit input: RX Clock used to sample LOAD, CE, INC
        .RX_CNTVALUEIN(RX_CNTVALUEIN), // 9-bit input: RX Counter value from device logic
        .RX_EN_VTC(RX_EN_VTC), // 1-bit input: RX Enable to keep stable delay over VT
        .RX_INC(RX_INC), // 1-bit input: RX Increment the current delay tap setting
        .RX_LOAD(RX_LOAD), // 1-bit input: RX Load the CNTVALUEIN tap setting
        .RX_RST(RX_RST), // 1-bit input: RX Asynchronous assert, synchronous deassert for
        // RXTX_BITSLICE ISERDES

        .RX_RST_DLY(RX_RST_DLY), // 1-bit input: RX Reset the internal DELAY value to DELAY_VALUE
        .T(T), // 1-bit input: Legacy T byte input from device logic
        .TBYTE_IN(TBYTE_IN), // 1-bit input: Byte group 3-state input from TX_BITSLICE_TRI
        .TX_BIT_CTRL_IN(TX_BIT_CTRL_IN), // 40-bit input: TX Input bus from BITSlice_CONTROL
        .TX_CE(TX_CE), // 1-bit input: Clock enable for ODELAY
        .TX_CLK(TX_CLK), // 1-bit input: TX Clock used to sample LOAD, CE, INC
        .TX_CNTVALUEIN(TX_CNTVALUEIN), // 9-bit input: TX Counter value from device logic
        .TX_EN_VTC(TX_EN_VTC), // 1-bit input: TX Enable to keep stable delay over VT
        .TX_INC(TX_INC), // 1-bit input: TX Increment the current delay tap setting
        .TX_LOAD(TX_LOAD), // 1-bit input: TX Load the CNTVALUEIN tap setting
        .TX_RST(TX_RST), // 1-bit input: TX Asynchronous assert, synchronous deassert for
        // RXTX_BITSLICE OSERDES

        .TX_RST_DLY(TX_RST_DLY) // 1-bit input: TX Reset the internal DELAY value to DELAY_VALUE
    );
    // End of RXTX_BITSLICE_inst instantiation

```

## For More Information

- See the *UltraScale Architecture SelectIO Resources User Guide* ([UG571](#)).
- See the [UltraScale Programmable Devices Documentation](#).

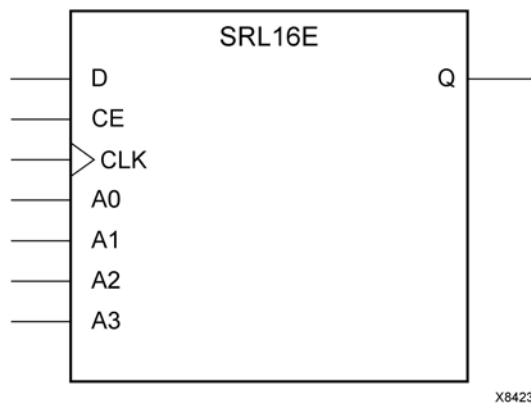
## SRL16E

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

PRIMITIVE\_GROUP: CLB

PRIMITIVE\_SUBGROUP: SRL

Families: UltraScale, UltraScale+



### Introduction

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

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

To create a fixed-depth shift register: Drive the A3 through A0 inputs with static values. The depth of the shift register can vary from 1 bit to 16 bits, as determined by the following formula:

$$\text{Depth} = (8 \times A3) + (4 \times A2) + (2 \times A1) + A0 + 1$$

If A3, A2, A1, and A0 are all zeroes (0000), the shift register is one bit deep. If they are all ones (1111), it is 16 bits deep.

To change the depth 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 depth of the shift register changes from 16 bits to 8 bits. Internally, the depth 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 zeroes (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 clock (CLK) transition. During subsequent 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 and retains current data within the shift register.

Two SLR16E components may be placed within the same LUT within a CLBM as long as they have the same clock, clock enable and depth selection address signals as well as the same IS\_CLK\_INVERTED attribute value. This allows up to 16 SRL16E components to be placed into a single CLB. Optionally, LUTNM or HLUTNMs may be placed on two SRL16E components to specify specific grouping within a LUT.

**NOTE:** When using SRLs with initialized values, you should use safe clock start-up techniques to ensure the initialized data is not corrupted upon completion of configuration. Refer to UG949: UltraFast Design Methodology Guide for details on controlling and synchronizing clock startup.

## 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
CE	Input	1	Active-High clock enable
CLK	Input	1	Shift register clock. Polarity is determined by the IS_CLK_INVERTED attribute.
D	Input	1	SRL data input
Q	Output	1	SRL data output
<b>Depth Selection:</b> The value placed on the A0 - A3 inputs specifies the shift register depth. <b>Depth = (8 x A3) + (4 x A2) + (2 x A1) + A0 + 1.</b>			
A0	Input	1	The value placed on the A0 - A3 inputs specifies the shift register depth. <b>Depth = (8 x A3) + (4 x A2) + (2 x A1) + A0 + 1.</b>
A1	Input	1	The value placed on the A0 - A3 inputs specifies the shift register depth. <b>Depth = (8 x A3) + (4 x A2) + (2 x A1) + A0 + 1.</b>
A2	Input	1	The value placed on the A0 - A3 inputs specifies the shift register depth. <b>Depth = (8 x A3) + (4 x A2) + (2 x A1) + A0 + 1.</b>
A3	Input	1	The value placed on the A0 - A3 inputs specifies the shift register depth. <b>Depth = (8 x A3) + (4 x A2) + (2 x A1) + A0 + 1.</b>

## Design Entry Method

Instantiation	Yes
Inference	Recommended
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	HEX	16'h0000 to 16'hffff	16'h0000	Specifies the initial contents in the shift register upon completion of configuration.
IS_CLK_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the clock pin (CLK) of this component. When set to 1 the active edge of the clock is the falling edge. If an external inverter is connected to this pin, the Vivado Design Suite will automatically set this attribute during the opt_design stage so that additional logic is not necessary for changing the clock polarity.

## VHDL 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 Look-Up Table (LUT)
--      UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

SRL16E_inst : SRL16E
generic map (
    INIT => X"0000",      -- Initial contents of shift register
    IS_CLK_INVERTED => '0' -- Optional inversion for CLK
)
port map (
    Q => Q,      -- 1-bit output: SRL Data
    CE => CE,    -- 1-bit input: Clock enable
    CLK => CLK,  -- 1-bit input: Clock
    D => D,      -- 1-bit input: SRL Data
    -- Depth Selection inputs: A0-A3 select SRL depth
    A0 => A0,
    A1 => A1,
    A2 => A2,
    A3 => A3
);

-- End of SRL16E_inst instantiation
    
```

## Verilog Instantiation Template

```

// SRL16E: 16-Bit Shift Register Look-Up Table (LUT)
//      UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

SRL16E #(
    .INIT(16'h0000),      // Initial contents of shift register
    .IS_CLK_INVERTED(1'b0) // Optional inversion for CLK
)
SRL16E_inst (
    .Q(Q),      // 1-bit output: SRL Data
    .CE(CE),    // 1-bit input: Clock enable
    .CLK(CLK),  // 1-bit input: Clock
    .D(D),      // 1-bit input: SRL Data
    // Depth Selection inputs: A0-A3 select SRL depth
    )
    
```

```
.A0 (A0),  
.A1 (A1),  
.A2 (A2),  
.A3 (A3)  
);  
  
// End of SRL16E_inst instantiation
```

## For More Information

- See the *UltraScale Architecture Configurable Logic Block User Guide* ([UG574](#)).
- See the [UltraScale Programmable Devices Documentation](#).

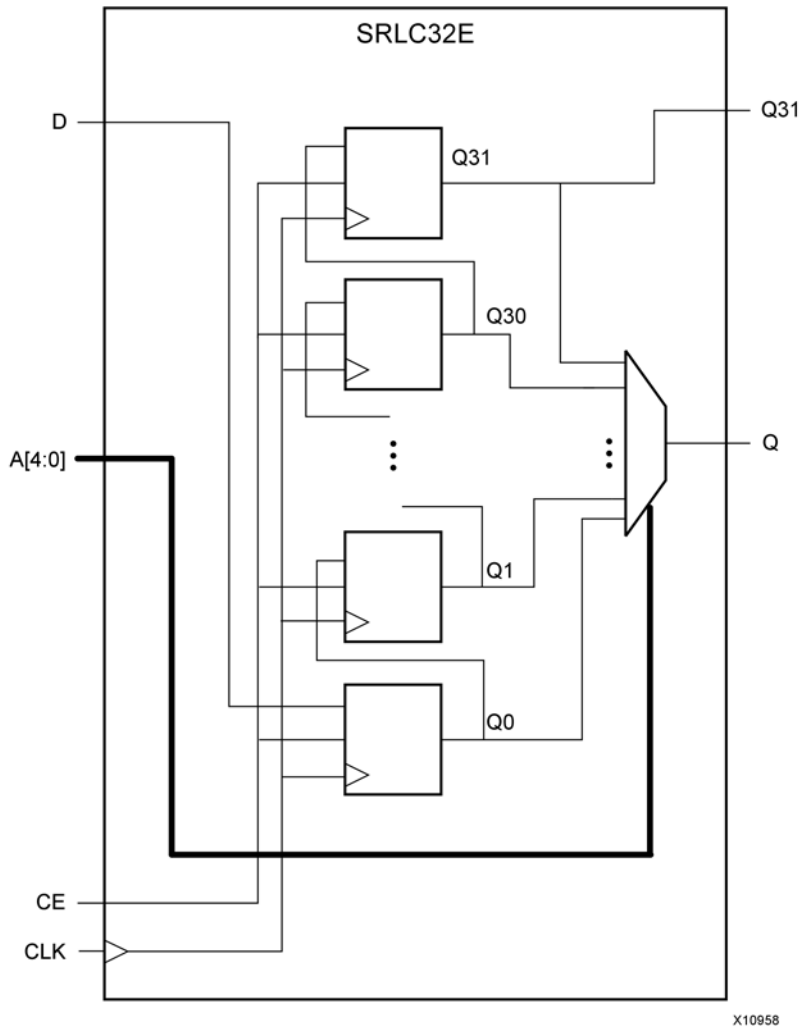
## SRLC32E

Primitive: 32-Bit Shift Register Look-Up Table (LUT)

PRIMITIVE\_GROUP: CLB

PRIMITIVE\_SUBGROUP: SRL

Families: UltraScale, UltraScale+



### Introduction

This design element is a shift register look-up table (LUT). The inputs A4, A3, A2, A1, and A0 select the depth of the shift register.

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

To create a fixed-depth shift register: Drive the A4 through A0 inputs with static values. The depth of the shift register can vary from 1 bit to 32 bits, as determined by the following formula:

$$\text{Depth} = (16 \times A4) + (8 \times A3) + (4 \times A2) + (2 \times A1) + A0 + 1$$



If A4, A3, A2, A1, and A0 are all zeroes (00000), the shift register is one bit deep. If they are all ones (11111), it is 32 bits deep.

To change the depth of the shift register dynamically: Change the values driving the A4 through A0 inputs. For example, if A3, A2, A1, and A0 are all ones (1111) and A4 toggles between a one (1) and a zero (0), the depth of the shift register changes from 32 bits to 16 bits. Internally, the depth of the shift register is always 32 bits and the input lines A4 through A0 select which of the 32 bits reach the output. The shift register LUT contents are initialized by assigning an eight-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 eight zeroes (00000000) 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 clock (CLK) transition. During subsequent 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 and retains current data within the shift register.

Two or more SRLC32E components may be cascaded to create deeper than 32-bit shift registers. To do so, connect the Q31 output of one SRLC32E component to the D input of another.

**NOTE:** When using SRLs with initialized values, you should use safe clock start-up techniques to ensure the initialized data is not corrupted upon completion of configuration. Refer to UG949: UltraFast Design Methodology Guide for details on controlling and synchronizing clock startup.

## Port Descriptions

Port	Direction	Width	Function
A<4:0>	Input	5	The value placed on the A0 - A3 inputs specifies the shift register depth. <b>Depth = (16 x A4) + (8 x A3) + (4 x A2) + (2 x A1) + A0 + 1.</b>
CE	Input	1	Active-High clock enable
CLK	Input	1	Shift register clock. Polarity is determined by the IS_CLK_INVERTED attribute.
D	Input	1	SRL data input
Q	Output	1	SRL data output
Q31	Output	1	SRL data output used to connect more than one SRLC32E component to form deeper than 32-bit shift registers.

## Design Entry Method

Instantiation	Yes
Inference	Recommended
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	HEX	Any 32-bit HEX value	All zeroes	Specifies the initial contents in the shift register upon completion of configuration.
IS_CLK_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the clock pin (CLK) of this component. When set to 1 the active edge of the clock is the falling edge. If an external inverter is connected to this pin, the Vivado Design Suite will automatically set this attribute during the opt_design stage so that additional logic is not necessary for changing the clock polarity.

## VHDL 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 Shift Register Look-Up Table (LUT)
-- UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

SRLC32E_inst : SRLC32E
generic map (
    INIT => X"00000000",    -- Initial contents of shift register
    IS_CLK_INVERTED => '0' -- Optional inversion for CLK
)
port map (
    Q => Q,        -- 1-bit output: SRL Data
    Q31 => Q31,    -- 1-bit output: SRL Cascade Data
    A => A,        -- 5-bit input: Selects SRL depth
    CE => CE,     -- 1-bit input: Clock enable
    CLK => CLK,   -- 1-bit input: Clock
    D => D        -- 1-bit input: SRL Data
);

-- End of SRLC32E_inst instantiation
    
```

## Verilog Instantiation Template

```

// SRLC32E: 32-Bit Shift Register Look-Up Table (LUT)
// UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

SRLC32E #(
    .INIT(32'h00000000),    // Initial contents of shift register
    .IS_CLK_INVERTED(1'b0) // Optional inversion for CLK
)
SRLC32E_inst (
    .Q(Q),        // 1-bit output: SRL Data
    .Q31(Q31),   // 1-bit output: SRL Cascade Data
    .A(A),       // 5-bit input: Selects SRL depth
    .CE(CE),     // 1-bit input: Clock enable
    .CLK(CLK),   // 1-bit input: Clock
    .D(D)        // 1-bit input: SRL Data
);
    
```

```
// End of SRLC32E_inst instantiation
```

## For More Information

- See the *UltraScale Architecture Configurable Logic Block User Guide* ([UG574](#)).
- See the [UltraScale Programmable Devices Documentation](#).

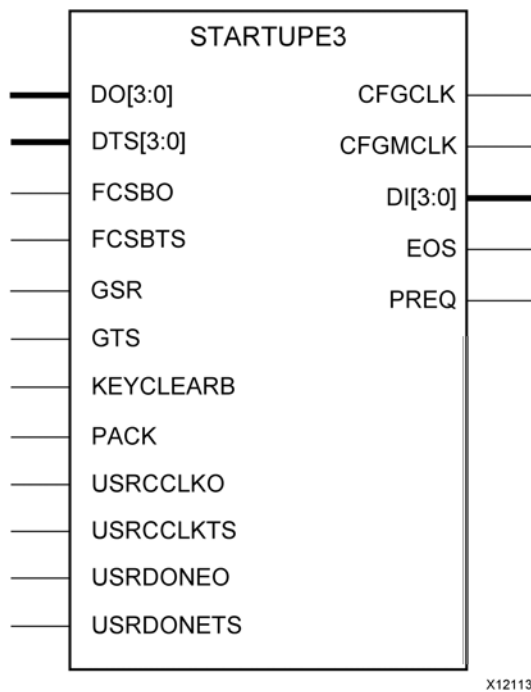
# STARTUPE3

Primitive: STARTUP Block

PRIMITIVE\_GROUP: CONFIGURATION

PRIMITIVE\_SUBGROUP: STARTUP

Families: UltraScale, UltraScale+



## Introduction

This design element is used to interface device pins and logic to the global asynchronous set/reset (GSR) signal, the global 3-state (GTS) dedicated routing or the internal configuration signals or a few of the dedicated configuration pins.

## Port Descriptions

Port	Direction	Width	Function
CFGCLK	Output	1	Configuration main clock output
CFGMCLK	Output	1	Configuration internal oscillator clock output
DI<3:0>	Output	4	Allow receiving on the D input pin
DO<3:0>	Input	4	Allows control of the D pin output
DTS<3:0>	Input	4	Allows tristate of the D pin
EOS	Output	1	Active-High output signal indicating the End Of Startup
FCSBO	Input	1	Controls the FCS_B pin for flash access
FCSBTS	Input	1	Tristate the FCS_B pin

Port	Direction	Width	Function
GSR	Input	1	Global Set/Reset input (GSR cannot be used for the port)
GTS	Input	1	Global 3-state input (GTS cannot be used for the port name)
KEYCLEARB	Input	1	Clear AES Decrypter Key input from Battery-Backed RAM (BBRAM)
PACK	Input	1	PROGRAM acknowledge input
PREQ	Output	1	PROGRAM request to fabric output
USRCCLKO	Input	1	User CCLK input
USRCCLKTS	Input	1	User CCLK 3-state enable input
USRDONEO	Input	1	User DONE pin output control
USRDONETS	Input	1	User DONE 3-state enable output

## Design Entry Method

Instantiation	Recommended
Inference	No
IP and IP Integrator Catalog	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
PROG_USR	STRING	"FALSE", "TRUE"	"FALSE"	Activate program event security feature. Requires encrypted bitstreams.
SIM_CCLK_FREQ	FLOAT(ns)	0.0 to 10.0	0.0	Set the Configuration Clock Frequency (ns) for simulation

## VHDL Instantiation Template

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- STARTUPE3: STARTUP Block
--           UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

STARTUPE3_inst : STARTUPE3
generic map (
    PROG_USR => "FALSE", -- Activate program event security feature. Requires encrypted bitstreams.
    SIM_CCLK_FREQ => 0.0 -- Set the Configuration Clock Frequency (ns) for simulation
)
port map (
    CFGCLK => CFGCLK, -- 1-bit output: Configuration main clock output
    CFGMCLK => CFGMCLK, -- 1-bit output: Configuration internal oscillator clock output
    DI => DI, -- 4-bit output: Allow receiving on the D input pin
    EOS => EOS, -- 1-bit output: Active-High output signal indicating the End Of Startup
    PREQ => PREQ, -- 1-bit output: PROGRAM request to fabric output
    DO => DO, -- 4-bit input: Allows control of the D pin output
    DTS => DTS, -- 4-bit input: Allows tristate of the D pin
    
```

```

FCSBO => FCSBO,          -- 1-bit input: Controls the FCS_B pin for flash access
FCSBTS => FCSBTS,       -- 1-bit input: Tristate the FCS_B pin
GSR => GSR,             -- 1-bit input: Global Set/Reset input (GSR cannot be used for the port)
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 (BDRAM)
PACK => PACK,           -- 1-bit input: PROGRAM acknowledge input
USRCLKO => USRCLKO,     -- 1-bit input: User CCLK input
USRCLKTS => USRCLKTS,   -- 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 STARTUPE3_inst instantiation
    
```

## Verilog Instantiation Template

```

// STARTUPE3: STARTUP Block
//           UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

STARTUPE3 #(
    .PROG_USR("FALSE"), // Activate program event security feature. Requires encrypted bitstreams.
    .SIM_CCLK_FREQ(0.0) // Set the Configuration Clock Frequency (ns) for simulation
)
STARTUPE3_inst (
    .CFGCLK(CFGCLK), // 1-bit output: Configuration main clock output
    .CFGMCLK(CFGMCLK), // 1-bit output: Configuration internal oscillator clock output
    .DI(DI), // 4-bit output: Allow receiving on the D input pin
    .EOS(EOS), // 1-bit output: Active-High output signal indicating the End Of Startup
    .PREQ(PREQ), // 1-bit output: PROGRAM request to fabric output
    .DO(DO), // 4-bit input: Allows control of the D pin output
    .DTS(DTS), // 4-bit input: Allows tristate of the D pin
    .FCSBO(FCSBO), // 1-bit input: Controls the FCS_B pin for flash access
    .FCSBTS(FCSBTS), // 1-bit input: Tristate the FCS_B pin
    .GSR(GSR), // 1-bit input: Global Set/Reset input (GSR cannot be used for the port)
    .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 (BDRAM)
    .PACK(PACK), // 1-bit input: PROGRAM acknowledge input
    .USRCLKO(USRCLKO), // 1-bit input: User CCLK input
    .USRCLKTS(USRCLKTS), // 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 STARTUPE3_inst instantiation
    
```

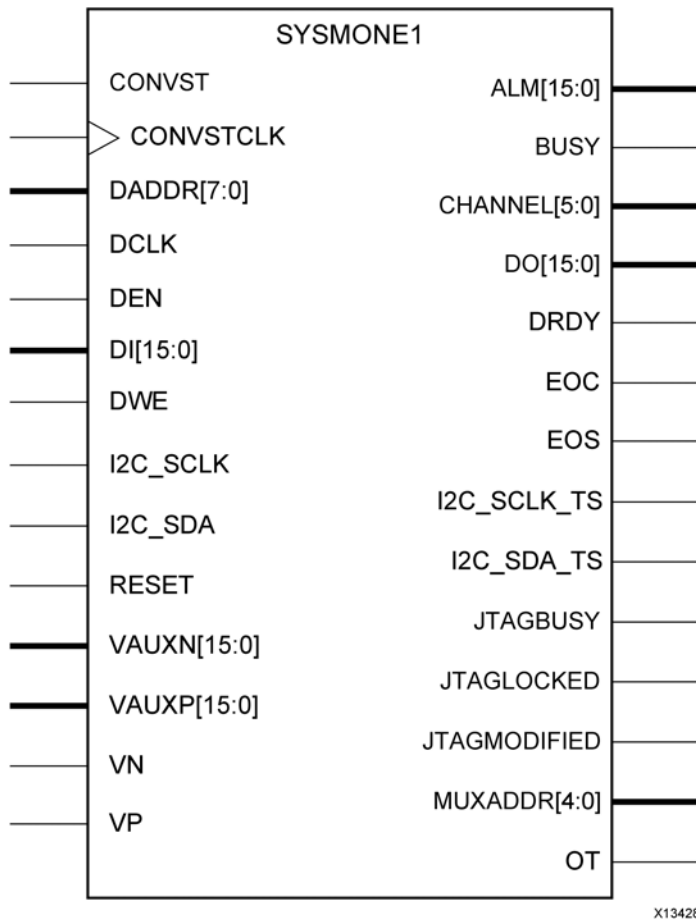
## For More Information

- See the *UltraScale Architecture Configuration User Guide* ([UG570](#)).
- See the [UltraScale Programmable Devices Documentation](#).

# SYSMONE1

Primitive: Xilinx Analog-to-Digital Converter and System Monitor

PRIMITIVE\_GROUP: **ADVANCED**  
 PRIMITIVE\_SUBGROUP: **SYSMON**  
 Families: UltraScale, UltraScale+



## Introduction

SYSMON provides an analog-to-digital conversion and associated monitoring capability. The core ADC comprises a 10-bit, 0.2 MSPS (Mega-sample-per-second) ADC, providing a general purpose analog interface for a range of applications. The ADC supports a range of operating modes and various analog input signal types e.g., unipolar, differential etc. The System Monitor also includes a number of on-chip sensors that support measurement of the on-chip power supply voltages and die temperature.

## Port Descriptions

Port	Direction	Width	Function
<b>ALARMS:</b> SYSMON alarm ports.			

Port	Direction	Width	Function
ALM<15:0>	Output	16	Output alarm for temperature, Vccint, Vccaux and Vccbram. ALM[0] - System Monitor temperature sensor alarm output. ALM[1] - System Monitor Vccint sensor alarm output. ALM[2] - System Monitor Vccaux sensor alarm output. ALM[3] - System Monitor Vccbram sensor alarm output. ALM[6:4] - Not defined. ALM[7] - Logic OR of bus ALM[6:0]. Can be used to flag occurrence of any alarm. ALM[11:8] - Alarms for the User Supplies 1-4. ALM[14:12] - Not defined ALM[15] - Logical OR of bus ALM[14:8] which can be used to flag any alarm in this group.
OT	Output	1	Over-Temperature alarm
<b>Auxiliary Analog-Input Pairs:</b> Sixteen auxiliary analog input pairs. In addition to the dedicated differential analog input, SYSMON can access 16 differential analog inputs by configuring digital I/O as analog inputs. These inputs can also be enabled preconfiguration via the JTAG port.			
VAUXN<15:0>	Input	16	N-side auxiliary analog input
VAUXP<15:0>	Input	16	P-side auxiliary analog input
<b>CONTROL and CLOCK:</b> SYSMON reset, conversion start and clock inputs.			
CONVST	Input	1	Convert start input. This input controls the sampling instant on the SYSMON(s) input and is only used in event mode timing. This input comes from the general-purpose interconnect in the device logic.
CONVSTCLK	Input	1	Convert start clock input. This input is connected to a clock net. Like CONVST, this input controls the sampling instant on the SYSMON(s) inputs and is only used in event mode timing. This input comes from the local clock distribution network in the device logic. Thus, for the best control over the sampling instant (delay and jitter), a global clock input can be used as the CONVST source.
RESET	Input	1	Reset signal for the SYSMON control logic.
<b>Dedicated Analog Input Pair:</b> One dedicated analog input pair. SYSMON has one pair of dedicated analog input pins that provide a differential analog input. If you are designing with SYSMON but not using the dedicated external channel of VP and VN, connect both VP and VN to analog ground.			
VN	Input	1	N-side analog input
VP	Input	1	P-side analog input
<b>Dynamic Reconfiguration Port (DRP):</b> Dynamic Reconfiguration Ports, ports used to access and control the System Monitor block.			
DADDR<7:0>	Input	8	Address bus for the dynamic reconfiguration port.
DCLK	Input	1	Clock input for the dynamic reconfiguration port.
DEN	Input	1	Enable signal for the dynamic reconfiguration port.
DI<15:0>	Input	16	Input data bus for the dynamic reconfiguration port.
DO<15:0>	Output	16	Output data bus for dynamic reconfiguration port.
DRDY	Output	1	Data ready signal for the dynamic reconfiguration port.
DWE	Input	1	Write enable for the dynamic reconfiguration port.
<b>I2C Interface:</b> Ports used with the I2C DRP interface			



Port	Direction	Width	Function
I2C_SCLK	Input	1	Input for the I2C_SCLK. Required for DRP I2C interface. The I2C_SCLK_IN and the I2C_SCLK_TS ports must be connected to the dedicated I2C_SCLK package pin as described in the System Monitor User Guide.
I2C_SCLK_TS	Output	1	Output for the I2C_SCLK. Required for DRP I2C interface. The I2C_SCLK_IN and the I2C_SCLK_TS ports must be connected to the dedicated I2C_SCLK package pin as described in the System Monitor User Guide.
I2C_SDA	Input	1	Input for the I2C_SDA. Required for DRP I2C interface. The I2C_SDA_IN and the I2C_SDA_TS ports must be connected to the dedicated I2C_SDA package pin as described in the System Monitor User Guide.
I2C_SDA_TS	Output	1	Output for the I2C_SDA. Required for DRP I2C interface. The I2C_SDA_IN and the I2C_SDA_TS ports must be connected to the dedicated I2C_SDA package pin as described in the System Monitor User Guide.
<b>STATUS:</b> SYSMON status ports.			
BUSY	Output	1	SYSMON busy signal. This signal transitions High during an ADC conversion. This signal also transitions High for an extended period during an ADC or sensor calibration.
CHANNEL<5:0>	Output	6	Channel selection outputs. The SYSMON input MUX channel selection for the current ADC conversion is placed on these outputs at the end of an ADC conversion.
EOC	Output	1	End of Conversion signal. This signal transitions to an active-High at the end of an ADC conversion when the measurement is written to the status registers.
EOS	Output	1	End of Sequence. This signal transitions to active-High when the measurement data from the last channel in an automatic channel sequence is written to the status registers.
JTAGBUSY	Output	1	Used to indicate that a JTAG DRP transaction is in progress.
JTAGLOCKED	Output	1	Indicates that a DRP port lock request has been made by the JTAG interface. This signal is also used to indicate that the DRP is ready for access (when Low).
JTAGMODIFIED	Output	1	Used to indicate that a JTAG Write to the DRP has occurred.
MUXADDR<4:0>	Output	5	These outputs are used in external multiplexer mode. They indicate the address of the next channel in a sequence to be converted. They provide the channel address for an external multiplexer.

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	Recommended

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT_45	HEX	16'h0000 to 16'hffff	16'h0000	Analog Bus Register
<b>INIT_40 - INIT_44:</b> System Monitor configuration registers.				
INIT_40	HEX	16'h0000 to 16'hffff	16'h0000	Configuration register 0
INIT_41	HEX	16'h0000 to 16'hffff	16'h0000	Configuration register 1
INIT_42	HEX	16'h0000 to 16'hffff	16'h0000	Configuration register 2
INIT_43	HEX	16'h0000 to 16'hffff	16'h0000	Configuration register 3
INIT_44	HEX	16'h0000 to 16'hffff	16'h0000	Configuration register 4
<b>INIT_46 - INIT_4F:</b> Sequence registers used to program the System Monitor Channel				
INIT_4A	HEX	16'h0000 to 16'hffff	16'h0000	Sequence register 4
INIT_4B	HEX	16'h0000 to 16'hffff	16'h0000	Sequence register 5
INIT_4C	HEX	16'h0000 to 16'hffff	16'h0000	Sequence register 6
INIT_4D	HEX	16'h0000 to 16'hffff	16'h0000	Sequence register 7
INIT_4E	HEX	16'h0000 to 16'hffff	16'h0000	Sequence register 8
INIT_4F	HEX	16'h0000 to 16'hffff	16'h0000	Sequence register 9
INIT_46	HEX	16'h0000 to 16'hffff	16'h0000	Sequence register 0
INIT_47	HEX	16'h0000 to 16'hffff	16'h0000	Sequence register 1
INIT_48	HEX	16'h0000 to 16'hffff	16'h0000	Sequence register 2
INIT_49	HEX	16'h0000 to 16'hffff	16'h0000	Sequence register 3
<b>INIT_50 - INIT_5F:</b> Alarm threshold registers for the System Monitor alarm function.				
INIT_5A	HEX	16'h0000 to 16'hffff	16'h0000	Alarm limit register 10
INIT_5B	HEX	16'h0000 to 16'hffff	16'h0000	Alarm limit register 11
INIT_5C	HEX	16'h0000 to 16'hffff	16'h0000	Alarm limit register 12
INIT_5D	HEX	16'h0000 to 16'hffff	16'h0000	Alarm limit register 13

Attribute	Type	Allowed Values	Default	Description
INIT_5E	HEX	16'h0000 to 16'hffff	16'h0000	Alarm limit register 14
INIT_5F	HEX	16'h0000 to 16'hffff	16'h0000	Alarm limit register 15
INIT_50	HEX	16'h0000 to 16'hffff	16'h0000	Alarm limit register 0
INIT_51	HEX	16'h0000 to 16'hffff	16'h0000	Alarm limit register 1
INIT_52	HEX	16'h0000 to 16'hffff	16'h0000	Alarm limit register 2
INIT_53	HEX	16'h0000 to 16'hffff	16'h0000	Alarm limit register 3
INIT_54	HEX	16'h0000 to 16'hffff	16'h0000	Alarm limit register 4
INIT_55	HEX	16'h0000 to 16'hffff	16'h0000	Alarm limit register 5
INIT_56	HEX	16'h0000 to 16'hffff	16'h0000	Alarm limit register 6
INIT_57	HEX	16'h0000 to 16'hffff	16'h0000	Alarm limit register 7
INIT_58	HEX	16'h0000 to 16'hffff	16'h0000	Alarm limit register 8
INIT_59	HEX	16'h0000 to 16'hffff	16'h0000	Alarm limit register 9
<b>INIT_60 - INIT_6F: User supply alarms</b>				
INIT_6A	HEX	16'h0000 to 16'hffff	16'h0000	User supply alarm register 10
INIT_6B	HEX	16'h0000 to 16'hffff	16'h0000	User supply alarm register 11
INIT_6C	HEX	16'h0000 to 16'hffff	16'h0000	User supply alarm register 12
INIT_6D	HEX	16'h0000 to 16'hffff	16'h0000	User supply alarm register 13
INIT_6E	HEX	16'h0000 to 16'hffff	16'h0000	User supply alarm register 14
INIT_6F	HEX	16'h0000 to 16'hffff	16'h0000	User supply alarm register 15
INIT_60	HEX	16'h0000 to 16'hffff	16'h0000	User supply alarm register 0
INIT_61	HEX	16'h0000 to 16'hffff	16'h0000	User supply alarm register 1
INIT_62	HEX	16'h0000 to 16'hffff	16'h0000	User supply alarm register 2
INIT_63	HEX	16'h0000 to 16'hffff	16'h0000	User supply alarm register 3

Attribute	Type	Allowed Values	Default	Description
INIT_64	HEX	16'h0000 to 16'hffff	16'h0000	User supply alarm register 4
INIT_65	HEX	16'h0000 to 16'hffff	16'h0000	User supply alarm register 5
INIT_66	HEX	16'h0000 to 16'hffff	16'h0000	User supply alarm register 6
INIT_67	HEX	16'h0000 to 16'hffff	16'h0000	User supply alarm register 7
INIT_68	HEX	16'h0000 to 16'hffff	16'h0000	User supply alarm register 8
INIT_69	HEX	16'h0000 to 16'hffff	16'h0000	User supply alarm register 9
<b>INIT_73 - INIT_77:</b> System Monitor Test registers not for customer usage				
INIT_73	HEX	16'h0000 to 16'hffff	16'h0000	System Monitor Test registers not for customer usage
INIT_74	HEX	16'h0000 to 16'hffff	16'h0000	System Monitor Test registers not for customer usage
INIT_75	HEX	16'h0000 to 16'hffff	16'h0000	System Monitor Test registers not for customer usage
INIT_76	HEX	16'h0000 to 16'hffff	16'h0000	System Monitor Test registers not for customer usage
INIT_77	HEX	16'h0000 to 16'hffff	16'h0000	System Monitor Test registers not for customer usage
<b>Not Defined:</b> Parameters not defined and reserved for future use.				
INIT_7A	HEX	16'h0000 to 16'hffff	16'h0000	Reserved for future use
INIT_7B	HEX	16'h0000 to 16'hffff	16'h0000	Reserved for future use
INIT_7C	HEX	16'h0000 to 16'hffff	16'h0000	Reserved for future use
INIT_7D	HEX	16'h0000 to 16'hffff	16'h0000	Reserved for future use
INIT_7E	HEX	16'h0000 to 16'hffff	16'h0000	Reserved for future use
INIT_7F	HEX	16'h0000 to 16'hffff	16'h0000	Reserved for future use
INIT_70	HEX	16'h0000 to 16'hffff	16'h0000	Reserved for future use
INIT_71	HEX	16'h0000 to 16'hffff	16'h0000	Reserved for future use
INIT_72	HEX	16'h0000 to 16'hffff	16'h0000	Reserved for future use
INIT_78	HEX	16'h0000 to 16'hffff	16'h0000	Reserved for future use

Attribute	Type	Allowed Values	Default	Description
INIT_79	HEX	16'h0000 to 16'hffff	16'h0000	Reserved for future use
<p><b>Programmable Inversion Attributes:</b> Indicates whether or not to use the optional inversions on specific pins of this component to change the active polarity of the pin function. When set to 1 on a clock pin, this components clocks on the negative edge. When set to 1 on other pins, it changes the function to behave active-Low rather than active-High. For pins that are buses, the bit-width of this attribute should match that of the bit-width of the associated pins and a binary value indicates which inverters to use and which to bypass. If an external inverter is specified on one of these associated pins, the Vivado Design Suite will automatically set this attribute during the opt_design stage so that additional logic is not necessary for changing the input polarity.</p>				
IS_CONVSTCLK_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the CONVSTCLK pin of this component.
IS_DCLK_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the DCLK pin of this component.
<p><b>Simulation attributes:</b> Specify the following attributes to allow for proper simulation of the System Monitor block.</p>				
SIM_MONITOR_FILE	STRING	String	"design.txt"	Specify the file name (and directory if different from simulation directory) of file containing analog voltage and temperature data for SYSMON simulation behavior.
<p><b>User Voltage Monitor:</b> SYSMON User voltage monitor attributes</p>				
SYSMON_VUSER0_BANK	DECIMAL	0 to 999	0	Specify the IO Bank number to be used with User0 for monitoring
SYSMON_VUSER0_MONITOR	STRING	String	"NONE"	Specify the voltage to monitor for User0
SYSMON_VUSER1_BANK	DECIMAL	0 to 999	0	Specify the IO Bank number to be used with User1 for monitoring
SYSMON_VUSER1_MONITOR	STRING	String	"NONE"	Specify the voltage to monitor for User1
SYSMON_VUSER2_BANK	DECIMAL	0 to 999	0	Specify the IO Bank number to be used with User2 for monitoring
SYSMON_VUSER2_MONITOR	STRING	String	"NONE"	Specify the voltage to monitor for User2
SYSMON_VUSER3_BANK	DECIMAL	0 to 999	0	Specify the IO Bank number to be used with User3 for monitoring
SYSMON_VUSER3_MONITOR	STRING	String	"NONE"	Specify the voltage to monitor for User3

## VHDL Instantiation Template

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

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

```

-- SYSMONE1: Xilinx Analog-to-Digital Converter and System Monitor
--           UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

SYSMONE1_inst : SYSMONE1
generic map (
  -- INIT_40 - INIT_44: SYSMON configuration registers
  INIT_40 => X"0000",
  INIT_41 => X"0000",
  INIT_42 => X"0000",
  INIT_43 => X"0000",
  INIT_44 => X"0000",
  INIT_45 => X"0000",          -- Analog Bus Register
  -- INIT_46 - INIT_4F: Sequence Registers
  INIT_46 => X"0000",
  INIT_47 => X"0000",
  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_5F: Alarm Limit 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",
  INIT_58 => X"0000",
  INIT_59 => X"0000",
  INIT_5A => X"0000",
  INIT_5B => X"0000",
  INIT_5C => X"0000",
  INIT_5D => X"0000",
  INIT_5E => X"0000",
  INIT_5F => X"0000",
  -- INIT_60 - INIT_6F: User Supply Alarms
  INIT_60 => X"0000",
  INIT_61 => X"0000",
  INIT_62 => X"0000",
  INIT_63 => X"0000",
  INIT_64 => X"0000",
  INIT_65 => X"0000",
  INIT_66 => X"0000",
  INIT_67 => X"0000",
  INIT_68 => X"0000",
  INIT_69 => X"0000",
  INIT_6A => X"0000",
  INIT_6B => X"0000",
  INIT_6C => X"0000",
  INIT_6D => X"0000",
  INIT_6E => X"0000",
  INIT_6F => X"0000",
  -- Programmable Inversion Attributes: Specifies the use of the built-in programmable inversion on
  -- specific pins
  IS_CONVSTCLK_INVERTED => '0',      -- Optional inversion for CONVSTCLK, 0-1
  IS_DCLK_INVERTED => '0',         -- Optional inversion for DCLK, 0-1
  -- Simulation attributes: Set for proper simulation behavior
  SIM_MONITOR_FILE => "design.txt", -- Analog simulation data file name
  -- User Voltage Monitor: SYSMON User voltage monitor
  SYSMON_VUSER0_BANK => 0,        -- Specify IO Bank for User0
  SYSMON_VUSER0_MONITOR => "NONE", -- Specify Voltage for User0
  SYSMON_VUSER1_BANK => 0,        -- Specify IO Bank for User1
  SYSMON_VUSER1_MONITOR => "NONE", -- Specify Voltage for User1
  SYSMON_VUSER2_BANK => 0,        -- Specify IO Bank for User2
  SYSMON_VUSER2_MONITOR => "NONE", -- Specify Voltage for User2

```

```

        SYSMON_VUSER3_MONITOR => "NONE"    -- Specify Voltage for User3
    )
    port map (
        -- ALARMS outputs: ALM, OT
        ALM => ALM,                        -- 16-bit output: Output alarm for temp, Vccint, Vccaux and Vccbram
        OT => OT,                          -- 1-bit output: Over-Temperature alarm
        -- Dynamic Reconfiguration Port (DRP) outputs: Dynamic Reconfiguration Ports
        DO => DO,                          -- 16-bit output: DRP output data bus
        DRDY => DRDY,                      -- 1-bit output: DRP data ready
        -- I2C Interface outputs: Ports used with the I2C DRP interface
        I2C_SCLK_TS => I2C_SCLK_TS,        -- 1-bit output: I2C_SCLK output port
        I2C_SDA_TS => I2C_SDA_TS,         -- 1-bit output: I2C_SDA_TS output port
        -- STATUS outputs: SYSMON status ports
        BUSY => BUSY,                      -- 1-bit output: System Monitor busy output
        CHANNEL => CHANNEL,                -- 6-bit output: Channel selection outputs
        EOC => EOC,                       -- 1-bit output: End of Conversion
        EOS => EOS,                       -- 1-bit output: End of Sequence
        JTAGBUSY => JTAGBUSY,              -- 1-bit output: JTAG DRP transaction in progress output
        JTAGLOCKED => JTAGLOCKED,          -- 1-bit output: JTAG requested DRP port lock
        JTAGMODIFIED => JTAGMODIFIED,      -- 1-bit output: JTAG Write to the DRP has occurred
        MUXADDR => MUXADDR,                -- 5-bit output: External MUX channel decode
        -- Auxiliary Analog-Input Pairs inputs: 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 inputs: Reset, conversion start and clock inputs
        CONVST => CONVST,                  -- 1-bit input: Convert start input
        CONVSTCLK => CONVSTCLK,            -- 1-bit input: Convert start input
        RESET => RESET,                    -- 1-bit input: Active-High reset
        -- Dedicated Analog Input Pair inputs: VP/VN
        VN => VN,                          -- 1-bit input: N-side analog input
        VP => VP,                          -- 1-bit input: P-side analog input
        -- Dynamic Reconfiguration Port (DRP) inputs: Dynamic Reconfiguration Ports
        DADDR => DADDR,                    -- 8-bit input: DRP address bus
        DCLK => DCLK,                      -- 1-bit input: DRP clock
        DEN => DEN,                        -- 1-bit input: DRP enable signal
        DI => DI,                          -- 16-bit input: DRP input data bus
        DWE => DWE,                        -- 1-bit input: DRP write enable
        -- I2C Interface inputs: Ports used with the I2C DRP interface
        I2C_SCLK => I2C_SCLK,              -- 1-bit input: I2C_SCLK input port
        I2C_SDA => I2C_SDA,                -- 1-bit input: I2C_SDA input port
    );

-- End of SYSMONE1_inst instantiation
    
```

## Verilog Instantiation Template

```

// SYSMONE1: Xilinx Analog-to-Digital Converter and System Monitor
// UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

SYSMONE1 #(
    // INIT_40 - INIT_44: SYSMON configuration registers
    .INIT_40(16'h0000),
    .INIT_41(16'h0000),
    .INIT_42(16'h0000),
    .INIT_43(16'h0000),
    .INIT_44(16'h0000),
    .INIT_45(16'h0000), // Analog Bus Register
    // INIT_46 - INIT_4F: Sequence Registers
    .INIT_46(16'h0000),
    .INIT_47(16'h0000),
    .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_5F: Alarm Limit 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),
.INIT_58(16'h0000),
.INIT_59(16'h0000),
.INIT_5A(16'h0000),
.INIT_5B(16'h0000),
.INIT_5C(16'h0000),
.INIT_5D(16'h0000),
.INIT_5E(16'h0000),
.INIT_5F(16'h0000),
// INIT_60 - INIT_6F: User Supply Alarms
.INIT_60(16'h0000),
.INIT_61(16'h0000),
.INIT_62(16'h0000),
.INIT_63(16'h0000),
.INIT_64(16'h0000),
.INIT_65(16'h0000),
.INIT_66(16'h0000),
.INIT_67(16'h0000),
.INIT_68(16'h0000),
.INIT_69(16'h0000),
.INIT_6A(16'h0000),
.INIT_6B(16'h0000),
.INIT_6C(16'h0000),
.INIT_6D(16'h0000),
.INIT_6E(16'h0000),
.INIT_6F(16'h0000),
// Programmable Inversion Attributes: Specifies the use of the built-in programmable inversion on
// specific pins
.IS_CONVSTCLK_INVERTED(1'b0), // Optional inversion for CONVSTCLK, 0-1
.IS_DCLK_INVERTED(1'b0), // Optional inversion for DCLK, 0-1
// Simulation attributes: Set for proper simulation behavior
.SIM_MONITOR_FILE("design.txt"), // Analog simulation data file name
// User Voltage Monitor: SYSMON User voltage monitor
.SYSMON_VUSER0_BANK(0), // Specify IO Bank for User0
.SYSMON_VUSER0_MONITOR("NONE"), // Specify Voltage for User0
.SYSMON_VUSER1_BANK(0), // Specify IO Bank for User1
.SYSMON_VUSER1_MONITOR("NONE"), // Specify Voltage for User1
.SYSMON_VUSER2_BANK(0), // Specify IO Bank for User2
.SYSMON_VUSER2_MONITOR("NONE"), // Specify Voltage for User2
.SYSMON_VUSER3_MONITOR("NONE") // Specify Voltage for User3
)
SYSMONE1_inst (
// ALARMS outputs: ALM, OT
.ALM(ALM), // 16-bit output: Output alarm for temp, Vccint, Vccaux and Vccbram
.OT(OT), // 1-bit output: Over-Temperature alarm
// Dynamic Reconfiguration Port (DRP) outputs: Dynamic Reconfiguration Ports
.DO(DO), // 16-bit output: DRP output data bus
.DRDY(DRDY), // 1-bit output: DRP data ready
// I2C Interface outputs: Ports used with the I2C DRP interface
.I2C_SCLK_TS(I2C_SCLK_TS), // 1-bit output: I2C_SCLK output port
.I2C_SDA_TS(I2C_SDA_TS), // 1-bit output: I2C_SDA_TS output port
// STATUS outputs: SYSMON status ports
.BUSY(BUSY), // 1-bit output: System Monitor busy output
.CHANNEL(CHANNEL), // 6-bit output: Channel selection outputs
.EOC(EOC), // 1-bit output: End of Conversion
.EOS(EOS), // 1-bit output: End of Sequence
.JTAGBUSY(JTAGBUSY), // 1-bit output: JTAG DRP transaction in progress output
.JTAGLOCKED(JTAGLOCKED), // 1-bit output: JTAG requested DRP port lock
.JTAGMODIFIED(JTAGMODIFIED), // 1-bit output: JTAG Write to the DRP has occurred
.MUXADDR(MUXADDR), // 5-bit output: External MUX channel decode
// Auxiliary Analog-Input Pairs inputs: 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 inputs: Reset, conversion start and clock inputs
.CONVST(CONVST), // 1-bit input: Convert start input

```



```

.CONVSTCLK(CONVSTCLK),          // 1-bit input: Convert start input
.RESET(RESET),                 // 1-bit input: Active-High reset
// Dedicated Analog Input Pair inputs: VP/VN
.VN(VN),                       // 1-bit input: N-side analog input
.VP(VP),                       // 1-bit input: P-side analog input
// Dynamic Reconfiguration Port (DRP) inputs: Dynamic Reconfiguration Ports
.DADDR(DADDR),                 // 8-bit input: DRP address bus
.DCLK(DCLK),                   // 1-bit input: DRP clock
.DEN(DEN),                     // 1-bit input: DRP enable signal
.DI(DI),                       // 16-bit input: DRP input data bus
.DWE(DWE),                     // 1-bit input: DRP write enable
// I2C Interface inputs: Ports used with the I2C DRP interface
.I2C_SCLK(I2C_SCLK),           // 1-bit input: I2C_SCLK input port
.I2C_SDA(I2C_SDA)              // 1-bit input: I2C_SDA input port
);

// End of SYSMONE1_inst instantiation

```

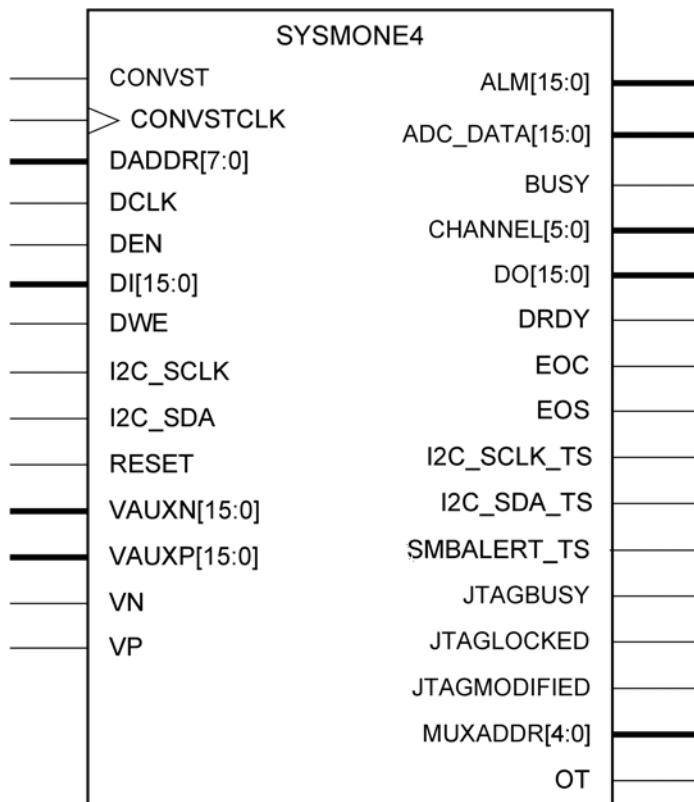
## For More Information

- See the *UltraScale Architecture System Monitor User Guide* ([UG580](#)).
- See the [UltraScale Programmable Devices Documentation](#).

# SYSMONE4

Primitive: Xilinx Analog-to-Digital Converter and System Monitor

PRIMITIVE\_GROUP: **ADVANCED**  
 PRIMITIVE\_SUBGROUP: **SYSMON**  
 Families: UltraScale+



X15115-102615

## Introduction

SYSMON provides an analog-to-digital conversion and associated monitoring capability. The ADC supports a range of operating modes and various analog input signal types e.g., unipolar, differential etc. The System Monitor also includes a number of on-chip sensors that support measurement of the on-chip power supply voltages and die temperature.

## Port Descriptions

Port	Direction	Width	Function
<b>ALARMS:</b> SYSMON alarm ports.			

Port	Direction	Width	Function
ALM<15:0>	Output	16	Output alarm for temperature, Vccint, Vccaux and Vccbram. ALM[0] - System Monitor temperature sensor alarm output. ALM[1] - System Monitor Vccint sensor alarm output. ALM[2] - System Monitor Vccaux sensor alarm output. ALM[3] - System Monitor Vccbram sensor alarm output. ALM[6:4] - Not defined. ALM[7] - Logic OR of bus ALM[6:0]. Can be used to flag occurrence of any alarm. ALM[11:8] - Alarms for the User Supplies 1-4. ALM[14:12] - Not defined ALM[15] - Logical OR of bus ALM[14:8] which can be used to flag any alarm in this group.
OT	Output	1	Over-Temperature alarm
<b>Auxiliary Analog-Input Pairs:</b> Sixteen auxiliary analog input pairs. In addition to the dedicated differential analog input, SYSMON can access 16 differential analog inputs by configuring digital I/O as analog inputs. These inputs can also be enabled preconfiguration via the JTAG port.			
VAUXN<15:0>	Input	16	N-side auxiliary analog input
VAUXP<15:0>	Input	16	P-side auxiliary analog input
<b>CONTROL and CLOCK:</b> SYSMON reset, conversion start and clock inputs.			
CONVST	Input	1	Convert start input. This input controls the sampling instant on the SYSMON(s) input and is only used in event mode timing. This input comes from the general-purpose interconnect in the device logic.
CONVSTCLK	Input	1	Convert start clock input. This input is connected to a clock net. Like CONVST, this input controls the sampling instant on the SYSMON(s) inputs and is only used in event mode timing. This input comes from the local clock distribution network in the device logic. Thus, for the best control over the sampling instant (delay and jitter), a global clock input can be used as the CONVST source.
RESET	Input	1	Reset signal for the SYSMON control logic.
<b>Dedicated Analog Input Pair:</b> One dedicated analog input pair. SYSMON has one pair of dedicated analog input pins that provide a differential analog input. If you are designing with SYSMON but not using the dedicated external channel of VP and VN, connect both VP and VN to analog ground.			
VN	Input	1	N-side analog input
VP	Input	1	P-side analog input
<b>Direct Data Out:</b> Direct data output. Measurement results updated every conversion (EOC). Use with channel			
ADC_DATA<15:0>	Output	16	Direct data output. Measurement results updated every conversion (EOC). Use with CHANNEL.
<b>Dynamic Reconfiguration Port (DRP):</b> Dynamic Reconfiguration Ports, ports used to access and control the System Monitor block.			
DADDR<7:0>	Input	8	Address bus for the dynamic reconfiguration port.
DCLK	Input	1	Clock input for the dynamic reconfiguration port.
DEN	Input	1	Enable signal for the dynamic reconfiguration port.
DI<15:0>	Input	16	Input data bus for the dynamic reconfiguration port.
DO<15:0>	Output	16	Output data bus for dynamic reconfiguration port.
DRDY	Output	1	Data ready signal for the dynamic reconfiguration port.

Port	Direction	Width	Function
DWE	Input	1	Write enable for the dynamic reconfiguration port.
<b>I2C Interface:</b> Ports used with the I2C DRP interface			
I2C_SCLK	Input	1	Input for the I2C_SCLK. Required for DRP I2C interface. The I2C_SCLK_IN and the I2C_SCLK_TS ports must be connected to the dedicated I2C_SCLK package pin as described in the System Monitor User Guide.
I2C_SCLK_TS	Output	1	Output for the I2C_SCLK. Required for DRP I2C interface. The I2C_SCLK_IN and the I2C_SCLK_TS ports must be connected to the dedicated I2C_SCLK package pin as described in the System Monitor User Guide.
I2C_SDA	Input	1	Input for the I2C_SDA. Required for DRP I2C interface. The I2C_SDA_IN and the I2C_SDA_TS ports must be connected to the dedicated I2C_SDA package pin as described in the System Monitor User Guide.
I2C_SDA_TS	Output	1	Output for the I2C_SDA. Required for DRP I2C interface. The I2C_SDA_IN and the I2C_SDA_TS ports must be connected to the dedicated I2C_SDA package pin as described in the System Monitor User Guide.
SMBALERT_TS	Output	1	Output control signal for SMBALERT. Connect to SMBALERT
<b>STATUS:</b> SYSMON status ports.			
BUSY	Output	1	SYSMON busy signal. This signal transitions High during an ADC conversion. This signal also transitions High for an extended period during an ADC or sensor calibration.
CHANNEL<5:0>	Output	6	Channel selection outputs. The SYSMON input MUX channel selection for the current ADC conversion is placed on these outputs at the end of an ADC conversion.
EOC	Output	1	End of Conversion signal. This signal transitions to an active-High at the end of an ADC conversion when the measurement is written to the status registers.
EOS	Output	1	End of Sequence. This signal transitions to active-High when the measurement data from the last channel in an automatic channel sequence is written to the status registers.
JTAGBUSY	Output	1	Used to indicate that a JTAG DRP transaction is in progress.
JTAGLOCKED	Output	1	Indicates that a DRP port lock request has been made by the JTAG interface. This signal is also used to indicate that the DRP is ready for access (when Low).
JTAGMODIFIED	Output	1	Used to indicate that a JTAG Write to the DRP has occurred.
MUXADDR<4:0>	Output	5	These outputs are used in external multiplexer mode. They indicate the address of the next channel in a sequence to be converted. They provide the channel address for an external multiplexer.

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	Recommended

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT_45	HEX	16'h0000 to 16'hffff	16'h0000	Analog Bus Register
<b>INIT_40 - INIT_44:</b> System Monitor configuration registers.				
INIT_40	HEX	16'h0000 to 16'hffff	16'h0000	Configuration register 0
INIT_41	HEX	16'h0000 to 16'hffff	16'h0000	Configuration register 1
INIT_42	HEX	16'h0000 to 16'hffff	16'h0000	Configuration register 2
INIT_43	HEX	16'h0000 to 16'hffff	16'h0000	Configuration register 3
INIT_44	HEX	16'h0000 to 16'hffff	16'h0000	Configuration register 4
<b>INIT_46 - INIT_4F:</b> Sequence registers used to program the System Monitor Channel				
INIT_4A	HEX	16'h0000 to 16'hffff	16'h0000	Sequence register 4
INIT_4B	HEX	16'h0000 to 16'hffff	16'h0000	Sequence register 5
INIT_4C	HEX	16'h0000 to 16'hffff	16'h0000	Sequence register 6
INIT_4D	HEX	16'h0000 to 16'hffff	16'h0000	Sequence register 7
INIT_4E	HEX	16'h0000 to 16'hffff	16'h0000	Sequence register 8
INIT_4F	HEX	16'h0000 to 16'hffff	16'h0000	Sequence register 9
INIT_46	HEX	16'h0000 to 16'hffff	16'h0000	Sequence register 0
INIT_47	HEX	16'h0000 to 16'hffff	16'h0000	Sequence register 1
INIT_48	HEX	16'h0000 to 16'hffff	16'h0000	Sequence register 2
INIT_49	HEX	16'h0000 to 16'hffff	16'h0000	Sequence register 3
<b>INIT_50 - INIT_5F:</b> Alarm threshold registers for the System Monitor alarm function.				
INIT_5A	HEX	16'h0000 to 16'hffff	16'h0000	Alarm limit register 10
INIT_5B	HEX	16'h0000 to 16'hffff	16'h0000	Alarm limit register 11
INIT_5C	HEX	16'h0000 to 16'hffff	16'h0000	Alarm limit register 12
INIT_5D	HEX	16'h0000 to 16'hffff	16'h0000	Alarm limit register 13
INIT_5E	HEX	16'h0000 to 16'hffff	16'h0000	Alarm limit register 14
INIT_5F	HEX	16'h0000 to 16'hffff	16'h0000	Alarm limit register 15
INIT_50	HEX	16'h0000 to 16'hffff	16'h0000	Alarm limit register 0
INIT_51	HEX	16'h0000 to 16'hffff	16'h0000	Alarm limit register 1
INIT_52	HEX	16'h0000 to 16'hffff	16'h0000	Alarm limit register 2
INIT_53	HEX	16'h0000 to 16'hffff	16'h0000	Alarm limit register 3
INIT_54	HEX	16'h0000 to 16'hffff	16'h0000	Alarm limit register 4
INIT_55	HEX	16'h0000 to 16'hffff	16'h0000	Alarm limit register 5
INIT_56	HEX	16'h0000 to 16'hffff	16'h0000	Alarm limit register 6
INIT_57	HEX	16'h0000 to 16'hffff	16'h0000	Alarm limit register 7
INIT_58	HEX	16'h0000 to 16'hffff	16'h0000	Alarm limit register 8
INIT_59	HEX	16'h0000 to 16'hffff	16'h0000	Alarm limit register 9

Attribute	Type	Allowed Values	Default	Description
<b>INIT_60 - INIT_6F: User supply alarms</b>				
INIT_6A	HEX	16'h0000 to 16'hffff	16'h0000	User supply alarm register 10
INIT_6B	HEX	16'h0000 to 16'hffff	16'h0000	User supply alarm register 11
INIT_6C	HEX	16'h0000 to 16'hffff	16'h0000	User supply alarm register 12
INIT_6D	HEX	16'h0000 to 16'hffff	16'h0000	User supply alarm register 13
INIT_6E	HEX	16'h0000 to 16'hffff	16'h0000	User supply alarm register 14
INIT_6F	HEX	16'h0000 to 16'hffff	16'h0000	User supply alarm register 15
INIT_60	HEX	16'h0000 to 16'hffff	16'h0000	User supply alarm register 0
INIT_61	HEX	16'h0000 to 16'hffff	16'h0000	User supply alarm register 1
INIT_62	HEX	16'h0000 to 16'hffff	16'h0000	User supply alarm register 2
INIT_63	HEX	16'h0000 to 16'hffff	16'h0000	User supply alarm register 3
INIT_64	HEX	16'h0000 to 16'hffff	16'h0000	User supply alarm register 4
INIT_65	HEX	16'h0000 to 16'hffff	16'h0000	User supply alarm register 5
INIT_66	HEX	16'h0000 to 16'hffff	16'h0000	User supply alarm register 6
INIT_67	HEX	16'h0000 to 16'hffff	16'h0000	User supply alarm register 7
INIT_68	HEX	16'h0000 to 16'hffff	16'h0000	User supply alarm register 8
INIT_69	HEX	16'h0000 to 16'hffff	16'h0000	User supply alarm register 9
<b>INIT_73 - INIT_77: System Monitor Test registers not for customer usage</b>				
INIT_73	HEX	16'h0000 to 16'hffff	16'h0000	System Monitor Test registers not for customer usage
INIT_74	HEX	16'h0000 to 16'hffff	16'h0000	System Monitor Test registers not for customer usage
INIT_75	HEX	16'h0000 to 16'hffff	16'h0000	System Monitor Test registers not for customer usage
INIT_76	HEX	16'h0000 to 16'hffff	16'h0000	System Monitor Test registers not for customer usage

Attribute	Type	Allowed Values	Default	Description
INIT_77	HEX	16'h0000 to 16'hffff	16'h0000	System Monitor Test registers not for customer usage
<b>Not Defined:</b> Parameters not defined and reserved for future use.				
INIT_7A	HEX	16'h0000 to 16'hffff	16'h0000	Reserved for future use
INIT_7B	HEX	16'h0000 to 16'hffff	16'h0000	Reserved for future use
INIT_7C	HEX	16'h0000 to 16'hffff	16'h0000	Reserved for future use
INIT_7D	HEX	16'h0000 to 16'hffff	16'h0000	Reserved for future use
INIT_7E	HEX	16'h0000 to 16'hffff	16'h0000	Reserved for future use
INIT_7F	HEX	16'h0000 to 16'hffff	16'h0000	Reserved for future use
INIT_70	HEX	16'h0000 to 16'hffff	16'h0000	Reserved for future use
INIT_71	HEX	16'h0000 to 16'hffff	16'h0000	Reserved for future use
INIT_72	HEX	16'h0000 to 16'hffff	16'h0000	Reserved for future use
INIT_78	HEX	16'h0000 to 16'hffff	16'h0000	Reserved for future use
INIT_79	HEX	16'h0000 to 16'hffff	16'h0000	Reserved for future use
<b>Primitive attributes:</b> Additional attributes on the primitive				
COMMON_N_SOURCE	HEX	16'h0000 to 16'hffff	16'hffff	Sets the auxiliary analog input that is used for the Common-N input. For example, if COMMON_N_SOURCE = 0h, VAUXN[0] is used.
<b>Programmable Inversion Attributes:</b> Indicates whether or not to use the optional inversions on specific pins of this component to change the active polarity of the pin function. When set to 1 on a clock pin, this components clocks on the negative edge. When set to 1 on other pins, it changes the function to behave active-Low rather than active-High. For pins that are buses, the bit-width of this attribute should match that of the bit-width of the associated pins and a binary value indicates which inverters to use and which to bypass. If an external inverter is specified on one of these associated pins, the Vivado Design Suite will automatically set this attribute during the opt_design stage so that additional logic is not necessary for changing the input polarity.				
IS_CONVSTCLK_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the CONVSTCLK pin of this component.
IS_DCLK_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether or not to use the optional inversion on the DCLK pin of this component.
<b>Simulation attributes:</b> Specify the following attributes to allow for proper simulation of the System Monitor block.				

Attribute	Type	Allowed Values	Default	Description
SIM_DEVICE	STRING	"ULTRASCALE_PLUS", "ULTRASCALE_PLUS_ES1", "ZYNQ_ULTRASCALE", "ZYNQ_ULTRASCALE_ES1"	"ULTRASCALE_PLUS"	Target device. Simulation models use SIM_DEVICE to determine the channels used for the default mode. Use ULTRASCALE_PLUS when using either Kintex UltraScale+ or Virtex UltraScale+ FPGAs.
SIM_MONITOR_FILE	STRING	String	"design.txt"	Specify the file name (and directory if different from simulation directory) of file containing analog voltage and temperature data for SYSMON simulation behavior.
<b>User Voltage Monitor:</b> SYSMON User voltage monitor attributes				
SYSMON_VUSER0_BANK	DECIMAL	0 to 999	0	Specify the IO Bank number to be used with User0 for monitoring
SYSMON_VUSER0_MONITOR	STRING	String	"NONE"	Specify the voltage to monitor for User0
SYSMON_VUSER1_BANK	DECIMAL	0 to 999	0	Specify the IO Bank number to be used with User1 for monitoring
SYSMON_VUSER1_MONITOR	STRING	String	"NONE"	Specify the voltage to monitor for User1
SYSMON_VUSER2_BANK	DECIMAL	0 to 999	0	Specify the IO Bank number to be used with User2 for monitoring
SYSMON_VUSER2_MONITOR	STRING	String	"NONE"	Specify the voltage to monitor for User2
SYSMON_VUSER3_BANK	DECIMAL	0 to 999	0	Specify the IO Bank number to be used with User3 for monitoring
SYSMON_VUSER3_MONITOR	STRING	String	"NONE"	Specify the voltage to monitor for User3

## VHDL Instantiation Template

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- SYSMONE4: Xilinx Analog-to-Digital Converter and System Monitor
--           UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

SYSMONE4_inst : SYSMONE4
generic map (
    -- INIT_40 - INIT_44: SYSMON configuration registers
    
```



```

INIT_40 => X"0000",
INIT_41 => X"0000",
INIT_42 => X"0000",
INIT_43 => X"0000",
INIT_44 => X"0000",
INIT_45 => X"0000",
-- INIT_46 - INIT_4F: Sequence Registers
INIT_46 => X"0000",
INIT_47 => X"0000",
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_5F: Alarm Limit 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",
INIT_58 => X"0000",
INIT_59 => X"0000",
INIT_5A => X"0000",
INIT_5B => X"0000",
INIT_5C => X"0000",
INIT_5D => X"0000",
INIT_5E => X"0000",
INIT_5F => X"0000",
-- INIT_60 - INIT_6F: User Supply Alarms
INIT_60 => X"0000",
INIT_61 => X"0000",
INIT_62 => X"0000",
INIT_63 => X"0000",
INIT_64 => X"0000",
INIT_65 => X"0000",
INIT_66 => X"0000",
INIT_67 => X"0000",
INIT_68 => X"0000",
INIT_69 => X"0000",
INIT_6A => X"0000",
INIT_6B => X"0000",
INIT_6C => X"0000",
INIT_6D => X"0000",
INIT_6E => X"0000",
INIT_6F => X"0000",
-- Primitive attributes: Primitive Attributes
COMMON_N_SOURCE => X"ffff", -- Sets the auxiliary analog input that is used for the Common-N input.
-- Programmable Inversion Attributes: Specifies the use of the built-in programmable inversion on
-- specific pins
IS_CONVSTCLK_INVERTED => '0', -- Optional inversion for CONVSTCLK, 0-1
IS_DCLK_INVERTED => '0', -- Optional inversion for DCLK, 0-1
-- Simulation attributes: Set for proper simulation behavior
SIM_DEVICE => "ULTRASCALE_PLUS", -- Sets the correct target device for simulation functionality.
SIM_MONITOR_FILE => "design.txt", -- Analog simulation data file name
-- User Voltage Monitor: SYSMON User voltage monitor
SYSMON_VUSER0_BANK => 0, -- Specify IO Bank for User0
SYSMON_VUSER0_MONITOR => "NONE", -- Specify Voltage for User0
SYSMON_VUSER1_BANK => 0, -- Specify IO Bank for User1
SYSMON_VUSER1_MONITOR => "NONE", -- Specify Voltage for User1
SYSMON_VUSER2_BANK => 0, -- Specify IO Bank for User2
SYSMON_VUSER2_MONITOR => "NONE", -- Specify Voltage for User2
SYSMON_VUSER3_MONITOR => "NONE" -- Specify Voltage for User3
)
port map (
-- ALARMS outputs: ALM, OT
ALM => ALM, -- 16-bit output: Output alarm for temp, Vccint, Vccaux and Vccbram

```

```

OT => OT,                -- 1-bit output: Over-Temperature alarm
-- Direct Data Out outputs: ADC_DATA
ADC_DATA => ADC_DATA,   -- 16-bit output: Direct Data Out
-- Dynamic Reconfiguration Port (DRP) outputs: Dynamic Reconfiguration Ports
DO => DO,                -- 16-bit output: DRP output data bus
DRDY => DRDY,           -- 1-bit output: DRP data ready
-- I2C Interface outputs: Ports used with the I2C DRP interface
I2C_SCLK_TS => I2C_SCLK_TS, -- 1-bit output: I2C_SCLK output port
I2C_SDA_TS => I2C_SDA_TS, -- 1-bit output: I2C_SDA_TS output port
SMBALERT_TS => SMBALERT_TS, -- 1-bit output: Output control signal for SMBALERT.
-- STATUS outputs: SYSMON status ports
BUSY => BUSY,           -- 1-bit output: System Monitor busy output
CHANNEL => CHANNEL,    -- 6-bit output: Channel selection outputs
EOC => EOC,             -- 1-bit output: End of Conversion
EOS => EOS,             -- 1-bit output: End of Sequence
JTAGBUSY => JTAGBUSY,   -- 1-bit output: JTAG DRP transaction in progress output
JTAGLOCKED => JTAGLOCKED, -- 1-bit output: JTAG requested DRP port lock
JTAGMODIFIED => JTAGMODIFIED, -- 1-bit output: JTAG Write to the DRP has occurred
MUXADDR => MUXADDR,    -- 5-bit output: External MUX channel decode
-- Auxiliary Analog-Input Pairs inputs: 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 inputs: Reset, conversion start and clock inputs
CONVST => CONVST,      -- 1-bit input: Convert start input
CONVSTCLK => CONVSTCLK, -- 1-bit input: Convert start input
RESET => RESET,        -- 1-bit input: Active-High reset
-- Dedicated Analog Input Pair inputs: VP/VN
VN => VN,               -- 1-bit input: N-side analog input
VP => VP,               -- 1-bit input: P-side analog input
-- Dynamic Reconfiguration Port (DRP) inputs: Dynamic Reconfiguration Ports
DADDR => DADDR,        -- 8-bit input: DRP address bus
DCLK => DCLK,          -- 1-bit input: DRP clock
DEN => DEN,            -- 1-bit input: DRP enable signal
DI => DI,              -- 16-bit input: DRP input data bus
DWE => DWE,            -- 1-bit input: DRP write enable
-- I2C Interface inputs: Ports used with the I2C DRP interface
I2C_SCLK => I2C_SCLK,  -- 1-bit input: I2C_SCLK input port
I2C_SDA => I2C_SDA     -- 1-bit input: I2C_SDA input port
);

-- End of SYSMONE4_inst instantiation
    
```

## Verilog Instantiation Template

```

// SYSMONE4: Xilinx Analog-to-Digital Converter and System Monitor
//           UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

SYSMONE4 #(
    // INIT_40 - INIT_44: SYSMON configuration registers
    .INIT_40(16'h0000),
    .INIT_41(16'h0000),
    .INIT_42(16'h0000),
    .INIT_43(16'h0000),
    .INIT_44(16'h0000),
    .INIT_45(16'h0000),           // Analog Bus Register
    // INIT_46 - INIT_4F: Sequence Registers
    .INIT_46(16'h0000),
    .INIT_47(16'h0000),
    .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_5F: Alarm Limit 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),
.INIT_58(16'h0000),
.INIT_59(16'h0000),
.INIT_5A(16'h0000),
.INIT_5B(16'h0000),
.INIT_5C(16'h0000),
.INIT_5D(16'h0000),
.INIT_5E(16'h0000),
.INIT_5F(16'h0000),
// INIT_60 - INIT_6F: User Supply Alarms
.INIT_60(16'h0000),
.INIT_61(16'h0000),
.INIT_62(16'h0000),
.INIT_63(16'h0000),
.INIT_64(16'h0000),
.INIT_65(16'h0000),
.INIT_66(16'h0000),
.INIT_67(16'h0000),
.INIT_68(16'h0000),
.INIT_69(16'h0000),
.INIT_6A(16'h0000),
.INIT_6B(16'h0000),
.INIT_6C(16'h0000),
.INIT_6D(16'h0000),
.INIT_6E(16'h0000),
.INIT_6F(16'h0000),
// Primitive attributes: Primitive Attributes
.COMMON_N_SOURCE(16'hffff), // Sets the auxiliary analog input that is used for the Common-N input.
// Programmable Inversion Attributes: Specifies the use of the built-in programmable inversion on
// specific pins
.IS_CONVSTCLK_INVERTED(1'b0), // Optional inversion for CONVSTCLK, 0-1
.IS_DCLK_INVERTED(1'b0), // Optional inversion for DCLK, 0-1
// Simulation attributes: Set for proper simulation behavior
.SIM_DEVICE("ULTRASCALE_PLUS"), // Sets the correct target device for simulation functionality.
.SIM_MONITOR_FILE("design.txt"), // Analog simulation data file name
// User Voltage Monitor: SYSMON User voltage monitor
.SYSMON_VUSER0_BANK(0), // Specify IO Bank for User0
.SYSMON_VUSER0_MONITOR("NONE"), // Specify Voltage for User0
.SYSMON_VUSER1_BANK(0), // Specify IO Bank for User1
.SYSMON_VUSER1_MONITOR("NONE"), // Specify Voltage for User1
.SYSMON_VUSER2_BANK(0), // Specify IO Bank for User2
.SYSMON_VUSER2_MONITOR("NONE"), // Specify Voltage for User2
.SYSMON_VUSER3_MONITOR("NONE") // Specify Voltage for User3
)
SYSMONE4 inst (
// ALARMS outputs: ALM, OT
.ALM(ALM), // 16-bit output: Output alarm for temp, Vccint, Vccaux and Vccbram
.OT(OT), // 1-bit output: Over-Temperature alarm
// Direct Data Out outputs: ADC_DATA
.ADC_DATA(ADC_DATA), // 16-bit output: Direct Data Out
// Dynamic Reconfiguration Port (DRP) outputs: Dynamic Reconfiguration Ports
.DO(DO), // 16-bit output: DRP output data bus
.DRDY(DRDY), // 1-bit output: DRP data ready
// I2C Interface outputs: Ports used with the I2C DRP interface
.I2C_SCLK_TS(I2C_SCLK_TS), // 1-bit output: I2C_SCLK output port
.I2C_SDA_TS(I2C_SDA_TS), // 1-bit output: I2C_SDA_TS output port
.SMBALERT_TS(SMBALERT_TS), // 1-bit output: Output control signal for SMBALERT.
// STATUS outputs: SYSMON status ports
.BUSY(BUSY), // 1-bit output: System Monitor busy output
.CHANNEL(CHANNEL), // 6-bit output: Channel selection outputs
.EOC(EOC), // 1-bit output: End of Conversion
.EOS(EOS), // 1-bit output: End of Sequence
.JTAGBUSY(JTAGBUSY), // 1-bit output: JTAG DRP transaction in progress output
.JTAGLOCKED(JTAGLOCKED), // 1-bit output: JTAG requested DRP port lock
.JTAGMODIFIED(JTAGMODIFIED), // 1-bit output: JTAG Write to the DRP has occurred
.MUXADDR(MUXADDR), // 5-bit output: External MUX channel decode
// Auxiliary Analog-Input Pairs inputs: 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 inputs: Reset, conversion start and clock inputs
.CONVST(CONVST),        // 1-bit input: Convert start input
.CONVSTCLK(CONVSTCLK),  // 1-bit input: Convert start input
.RESET(RESET),          // 1-bit input: Active-High reset
// Dedicated Analog Input Pair inputs: VP/VN
.VN(VN),                // 1-bit input: N-side analog input
.VP(VP),                // 1-bit input: P-side analog input
// Dynamic Reconfiguration Port (DRP) inputs: Dynamic Reconfiguration Ports
.DADDR(DADDR),          // 8-bit input: DRP address bus
.DCLK(DCLK),            // 1-bit input: DRP clock
.DEN(DEN),              // 1-bit input: DRP enable signal
.DI(DI),                // 16-bit input: DRP input data bus
.DWE(DWE),              // 1-bit input: DRP write enable
// I2C Interface inputs: Ports used with the I2C DRP interface
.I2C_SCLK(I2C_SCLK),    // 1-bit input: I2C_SCLK input port
.I2C_SDA(I2C_SDA)      // 1-bit input: I2C_SDA input port
);

// End of SYSMONE4_inst instantiation
    
```

## For More Information

- See the *UltraScale Architecture System Monitor User Guide* ([UG580](#)).
- See the [UltraScale Programmable Devices Documentation](#).

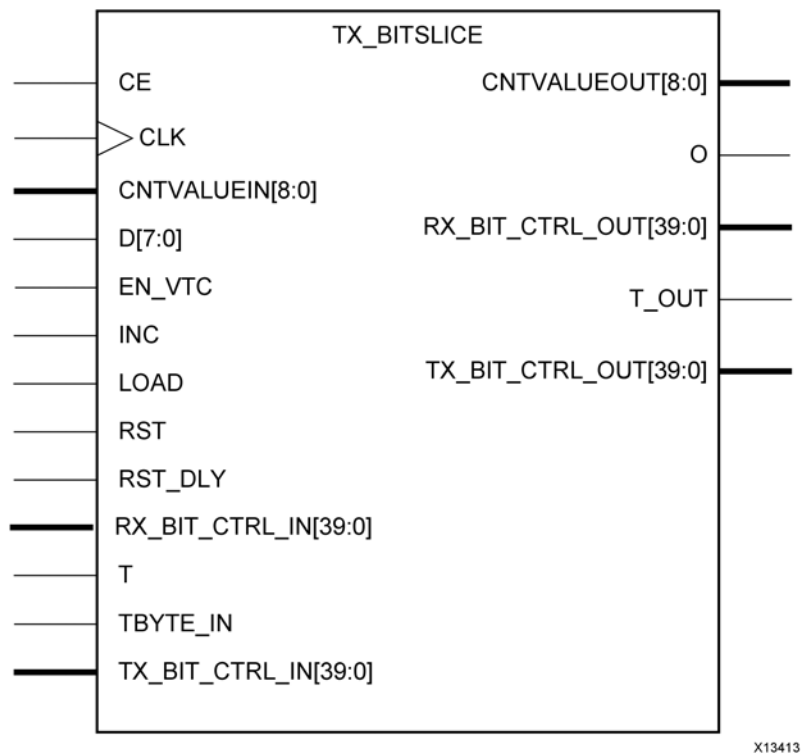
# TX\_BITSLICE

Primitive: TX\_BITSLICE for output using Native Mode

PRIMITIVE\_GROUP: I/O

PRIMITIVE\_SUBGROUP: BITSlice

Families: UltraScale, UltraScale+



## Introduction

In native mode, the TX\_BITSLICE contains serialization logic and a 512-tap output delay (ODELAY) that can be continuously adjusted for VT variation. The TX\_BITSLICE contains serialization logic for 4:1 or 8:1 serialization.

## Port Descriptions

Port	Direction	Width	Function
CE	Input	1	Clock enable for TX_BITSLICE ODELAY register clock
CLK	Input	1	Clock used to sample LOAD, CE, INC
CNTVALUEIN<8:0>	Input	9	Counter value from internal device logic for tap value to be loaded dynamically
CNTVALUEOUT<8:0>	Output	9	Counter value to going the internal device logic for monitoring tap value
D<7:0>	Input	8	Data from device logic

Port	Direction	Width	Function
EN_VTC	Input	1	Enable to keep stable delay over VT when set to HIGH. VT compensation disabled when set to LOW.
INC	Input	1	Increment the current delay tap setting
LOAD	Input	1	Load the CNTVALUEIN tap setting
O	Output	1	Serialized output going to output buffer
RST	Input	1	Asynchronous assert, synchronous deassert for TX_BITSLICE OSERDES
RST_DLY	Input	1	Reset the internal DELAY value to DELAY_VALUE
RX_BIT_CTRL_IN<39:0>	Input	40	Input bus from BITSlice_CONTROL
RX_BIT_CTRL_OUT<39:0>	Output	40	Output bus to BITSlice_CONTROL
T	Input	1	Legacy T byte input from device logic
T_OUT	Output	1	Byte group 3-state output
TBYTE_IN	Input	1	Byte group 3-state input from TX_BITSLICE_TRI
TX_BIT_CTRL_IN<39:0>	Input	40	Input bus from BITSlice_CONTROL
TX_BIT_CTRL_OUT<39:0>	Output	40	Output bus to BITSlice_CONTROL

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	Yes

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
DATA_WIDTH	DECIMAL	8, 4	8	Defines the parallel data input width. Legal values are 4 and 8.
DELAY_FORMAT	STRING	"TIME", "COUNT"	"TIME"	Sets the units of DELAY_VALUE of the TX_BITSLICE ODELAY. Use TIME when DELAY_TYPE is FIXED. Use COUNT when DELAY_TYPE is VARIABLE or VAR_LOAD. <ul style="list-style-type: none"> <li>"TIME" : TX_BITSLICE ODELAY DELAY_VALUE is specified in ps</li> <li>"COUNT" : TX_BITSLICE ODELAY DELAY_VALUE is specified in taps</li> </ul>

Attribute	Type	Allowed Values	Default	Description
DELAY_TYPE	STRING	"FIXED", "VAR_LOAD", "VARIABLE"	"FIXED"	Sets the type of tap delay line. <ul style="list-style-type: none"> <li>"FIXED" - Sets a static delay value</li> <li>"VARIABLE" - Dynamically adjust (increment/decrement) delay value</li> <li>"VAR_LOAD" - Dynamically loads tap values</li> </ul>
DELAY_VALUE	DECIMAL	0 to 1250	0	Specifies the fixed delay in ps when using FIXED DELAY_TYPE. Specifies value upon reset when using VARIABLE or VAR_LOAD.
ENABLE_PRE_EMPHASIS	STRING	"FALSE", "TRUE"	"FALSE"	Used in conjunction with IOB to Enable the pre-emphasis.
INIT	BINARY	1'b1, 1'b0	1'b1	Defines initial O value
IS_CLK_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether the clock CLK pin is active-high or active-low.
IS_RST_DLY_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether the reset RST_DLY pin is active-high or active-low.
IS_RST_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether the reset RST pin is active-high or active-low.
OUTPUT_PHASE_90	STRING	"FALSE", "TRUE"	"FALSE"	Delays the output phase by 90-degrees
REFCLK_FREQUENCY	1 significant digit FLOAT	200.0 to 2667.0	300.0	Specification of the reference clock frequency in MHz
SIM_DEVICE	STRING	"ULTRASCALE", "ULTRASCALE_PLUS_ES1"	"ULTRASCALE"	Set the device version
TBYTE_CTL	STRING	"TBYTE_IN", "T"	"TBYTE_IN"	Select between T and TBYTE_IN inputs in OSERDES mode only
UPDATE_MODE	STRING	"ASYNC", "MANUAL", "SYNC"	"ASYNC"	<ul style="list-style-type: none"> <li>"ASYNC": Updates are increments or decrements to the delay value independent of the data being received</li> <li>"SYNC": Updates require that input data transitions to synchronously update the delay with the data input edges</li> <li>"MANUAL": Updates take effect when both LOAD and CE are asserted after the LOAD and CNTVALUEIN signals are used to load the new CNTVALUE</li> </ul>

## VHDL Instantiation Template

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- TX_BITSLICE: TX_BITSLICE for output using Native Mode
-- UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

TX_BITSLICE_inst : TX_BITSLICE
generic map (
    DATA_WIDTH => 8,                -- Parallel data input width (4-8)
    DELAY_FORMAT => "TIME",          -- Units of the DELAY_VALUE (COUNT, TIME)
    DELAY_TYPE => "FIXED",          -- Set the type of tap delay line (FIXED, VARIABLE, VAR_LOAD)
    DELAY_VALUE => 0,                -- Output delay value setting
    ENABLE_PRE_EMPHASIS => "FALSE", -- Enable the pre-emphasis
    INIT => '1',                      -- Defines initial 0 value
    IS_CLK_INVERTED => '0',          -- Optional inversion for CLK
    IS_RST_DLY_INVERTED => '0',      -- Optional inversion for RST_DLY
    IS_RST_INVERTED => '0',         -- Optional inversion for RST
    OUTPUT_PHASE_90 => "FALSE",     -- Delays the output phase by 90-degrees
    REFCLK_FREQUENCY => 300.0,      -- Specification of the reference clock frequency in MHz (200.0-2667.0)
    SIM_DEVICE => "ULTRASCALE",     -- Set the device version (ULTRASCALE, ULTRASCALE_PLUS_ES1)
    TBYTE_CTL => "TBYTE_IN",        -- Select between T and TBYTE_IN inputs
    UPDATE_MODE => "ASYNCR",        -- Determines when updates to the delay will take effect (ASYNCR, MANUAL,
    -- SYNC)
)
port map (
    CNTVALUEOUT => CNTVALUEOUT,     -- 9-bit output: Counter value to device logic
    O => O,                          -- 1-bit output: Serialized output going to output buffer
    RX_BIT_CTRL_OUT => RX_BIT_CTRL_OUT, -- 40-bit output: Output bus to BITSlice_CONTROL
    TX_BIT_CTRL_OUT => TX_BIT_CTRL_OUT, -- 40-bit output: Output bus to BITSlice_CONTROL
    T_OUT => T_OUT,                  -- 1-bit output: Byte group 3-state output
    CE => CE,                        -- 1-bit input: Clock enable for ODELAY
    CLK => CLK,                      -- 1-bit input: Clock used to sample LOAD, CE, INC
    CNTVALUEIN => CNTVALUEIN,        -- 9-bit input: Counter value from device logic
    D => D,                          -- 8-bit input: Data from device logic
    EN_VTC => EN_VTC,                -- 1-bit input: Enable to keep stable delay over VT
    INC => INC,                      -- 1-bit input: Increment the current delay tap setting
    LOAD => LOAD,                    -- 1-bit input: Load the CNTVALUEIN tap setting
    RST => RST,                      -- 1-bit input: Asynchronous assert, synchronous deassert for
    -- TX_BITSLICE OSERDES

    RST_DLY => RST_DLY,              -- 1-bit input: Reset the internal DELAY value to DELAY_VALUE
    RX_BIT_CTRL_IN => RX_BIT_CTRL_IN, -- 40-bit input: Input bus from BITSlice_CONTROL
    T => T,                          -- 1-bit input: Legacy T byte input from device logic
    TBYTE_IN => TBYTE_IN,            -- 1-bit input: Byte group 3-state input from TX_BITSLICE_TRI
    TX_BIT_CTRL_IN => TX_BIT_CTRL_IN -- 40-bit input: Input bus from BITSlice_CONTROL
);

-- End of TX_BITSLICE_inst instantiation
    
```



## Verilog Instantiation Template

```
// TX_BITSLICE: TX_BITSLICE for output using Native Mode
// UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

TX_BITSLICE #(
    .DATA_WIDTH(8), // Parallel data input width (4-8)
    .DELAY_FORMAT("TIME"), // Units of the DELAY_VALUE (COUNT, TIME)
    .DELAY_TYPE("FIXED"), // Set the type of tap delay line (FIXED, VARIABLE, VAR_LOAD)
    .DELAY_VALUE(0), // Output delay value setting
    .ENABLE_PRE_EMPHASIS("FALSE"), // Enable the pre-emphasis
    .INIT(1'b1), // Defines initial 0 value
    .IS_CLK_INVERTED(1'b0), // Optional inversion for CLK
    .IS_RST_DLY_INVERTED(1'b0), // Optional inversion for RST_DLY
    .IS_RST_INVERTED(1'b0), // Optional inversion for RST
    .OUTPUT_PHASE_90("FALSE"), // Delays the output phase by 90-degrees
    .REFCLK_FREQUENCY(300.0), // Specification of the reference clock frequency in MHz (200.0-2667.0)
    .SIM_DEVICE("ULTRASCALE"), // Set the device version (ULTRASCALE, ULTRASCALE_PLUS_ES1)
    .TBYTE_CTL("TBYTE_IN"), // Select between T and TBYTE_IN inputs
    .UPDATE_MODE("ASYNC") // Determines when updates to the delay will take effect (ASYNC, MANUAL,
    // SYNC)
)
TX_BITSLICE_inst (
    .CNTVALUEOUT(CNTVALUEOUT), // 9-bit output: Counter value to device logic
    .O(O), // 1-bit output: Serialized output going to output buffer
    .RX_BIT_CTRL_OUT(RX_BIT_CTRL_OUT), // 40-bit output: Output bus to BITSlice_CONTROL
    .TX_BIT_CTRL_OUT(TX_BIT_CTRL_OUT), // 40-bit output: Output bus to BITSlice_CONTROL
    .T_OUT(T_OUT), // 1-bit output: Byte group 3-state output
    .CE(CE), // 1-bit input: Clock enable for ODELAY
    .CLK(CLK), // 1-bit input: Clock used to sample LOAD, CE, INC
    .CNTVALUEIN(CNTVALUEIN), // 9-bit input: Counter value from device logic
    .D(D), // 8-bit input: Data from device logic
    .EN_VTC(EN_VTC), // 1-bit input: Enable to keep stable delay over VT
    .INC(INC), // 1-bit input: Increment the current delay tap setting
    .LOAD(LOAD), // 1-bit input: Load the CNTVALUEIN tap setting
    .RST(RST), // 1-bit input: Asynchronous assert, synchronous deassert for
    // TX_BITSLICE OSERDES

    .RST_DLY(RST_DLY), // 1-bit input: Reset the internal DELAY value to DELAY_VALUE
    .RX_BIT_CTRL_IN(RX_BIT_CTRL_IN), // 40-bit input: Input bus from BITSlice_CONTROL
    .T(T), // 1-bit input: Legacy T byte input from device logic
    .TBYTE_IN(TBYTE_IN), // 1-bit input: Byte group 3-state input from TX_BITSLICE_TRI
    .TX_BIT_CTRL_IN(TX_BIT_CTRL_IN) // 40-bit input: Input bus from BITSlice_CONTROL
);

// End of TX_BITSLICE_inst instantiation
```

## For More Information

- See the *UltraScale Architecture SelectIO Resources User Guide* ([UG571](#)).
- See the [UltraScale Programmable Devices Documentation](#).

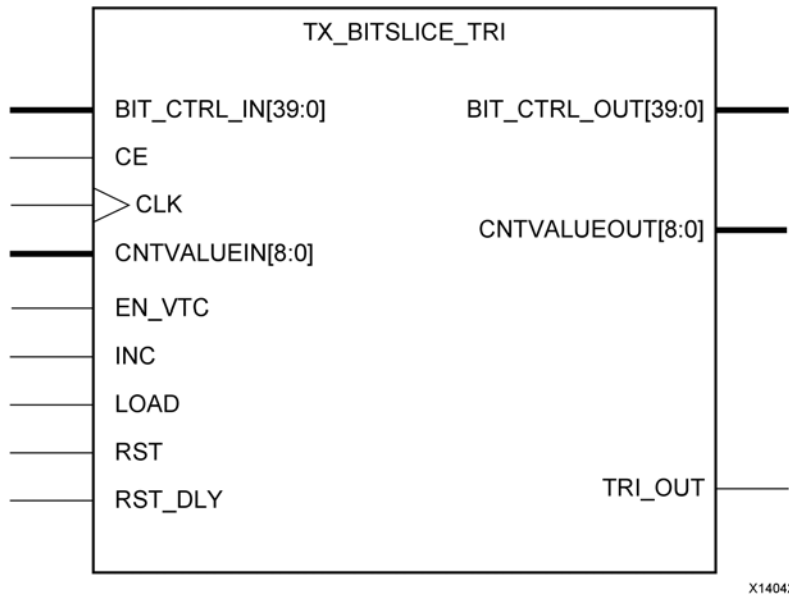
# TX\_BITSLICE\_TRI

Primitive: TX\_BITSLICE\_TRI for tristate using Native Mode

PRIMITIVE\_GROUP: I/O

PRIMITIVE\_SUBGROUP: BITSlice

Families: UltraScale, UltraScale+



## Introduction

In native mode, the TX\_BITSLICE\_TRI provides the ability to tristate bitslices within a nibble. The TX\_BITSLICE\_TRI also contains a 512-tap output delay element (ODELAY) with a calibrated tap resolution.

## Port Descriptions

Port	Direction	Width	Function
BIT_CTRL_IN<39:0>	Input	40	Input bus from BITSlice_CONTROL
BIT_CTRL_OUT<39:0>	Output	40	Output bus to BITSlice_CONTROL
CE	Input	1	Active high enable increment/decrement input
CLK	Input	1	Clock input
CNTVALUEIN<8:0>	Input	9	Counter value from device logic for dynamically loadable tap value input
CNTVALUEOUT<8:0>	Output	9	Counter value to going the internal device logic for monitoring tap value
EN_VTC	Input	1	Enable to keep stable delay over VT when set to HIGH. VT compensation disabled when set to LOW.
INC	Input	1	Increment the current delay tap setting

Port	Direction	Width	Function
LOAD	Input	1	Load the CNTVALUEIN tap setting
RST	Input	1	Asynchronous assert, synchronous deassert
RST_DLY	Input	1	Reset the internal DELAY value to DELAY_VALUE
TRI_OUT	Output	1	Output to the TBYTE_IN pins of the bitslices

## Design Entry Method

Instantiation	Yes
Inference	No
IP and IP Integrator Catalog	Yes

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
DATA_WIDTH	DECIMAL	8, 4	8	Defines the parallel data input width. Legal values are 4 and 8.
DELAY_FORMAT	STRING	"TIME", "COUNT"	"TIME"	Sets the units of DELAY_VALUE of the TX_BITSLICE_TRI ODELAY. Use TIME when DELAY_TYPE is FIXED. Use COUNT when DELAY_TYPE is VARIABLE or VAR_LOAD. <ul style="list-style-type: none"> <li>"TIME" : TX_BITSLICE_TRI ODELAY DELAY_VALUE is specified in ps</li> <li>"COUNT" : TX_BITSLICE_TRI ODELAY DELAY_VALUE is specified in taps</li> </ul>
DELAY_TYPE	STRING	"FIXED", "VAR_LOAD", "VARIABLE"	"FIXED"	Sets the type of tap delay line. <ul style="list-style-type: none"> <li>"FIXED" - Sets a static delay value</li> <li>"VARIABLE" - Dynamically adjust (increment/decrement) delay value</li> <li>"VAR_LOAD" - Dynamically loads tap values</li> </ul>
DELAY_VALUE	DECIMAL	0 to 1250	0	Specifies the fixed delay in ps when using FIXED DELAY_TYPE. Specifies value upon reset when using VARIABLE or VAR_LOAD.
INIT	BINARY	1'b1, 1'b0	1'b1	Defines initial O value
IS_CLK_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether the clock CLK pin is active-high or active-low.
IS_RST_DLY_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether the reset RST_DLY pin is active-high or active-low.

Attribute	Type	Allowed Values	Default	Description
IS_RST_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Specifies whether the reset RST pin is active-high or active-low.
OUTPUT_PHASE_90	STRING	"FALSE", "TRUE"	"FALSE"	Delays the output phase by 90-degrees
REFCLK_FREQUENCY	1 significant digit FLOAT	200.0 to 2667.0	300.0	Specification of the reference clock frequency in MHz
SIM_DEVICE	STRING	"ULTRASCALE", "ULTRASCALE_PLUS_ES1"	"ULTRASCALE"	Set the device version
UPDATE_MODE	STRING	"ASYNC", "MANUAL", "SYNC"	"ASYNC"	<ul style="list-style-type: none"> <li>"ASYNC": Updates are increments or decrements to the delay value independent of the data being received</li> <li>"SYNC": Updates require that input data transitions to synchronously update the delay with the data input edges</li> <li>"MANUAL": Updates take effect when both LOAD and CE are asserted after the LOAD and CNTVALUEIN signals are used to load the new CNTVALUE</li> </ul>

## VHDL Instantiation Template

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- TX_BITSLICE_TRI: TX_BITSLICE_TRI for tristate using Native Mode
-- UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

TX_BITSLICE_TRI_inst : TX_BITSLICE_TRI
generic map (
    DATA_WIDTH => 8,           -- Parallel data input width (4-8)
    DELAY_FORMAT => "TIME",     -- Units of the DELAY_VALUE (COUNT, TIME)
    DELAY_TYPE => "FIXED",      -- Set the type of tap delay line (FIXED, VARIABLE, VAR_LOAD)
    DELAY_VALUE => 0,           -- Output delay value setting
    INIT => '1',                -- Defines initial 0 value
    IS_CLK_INVERTED => '0',     -- Optional inversion for CLK
    IS_RST_DLY_INVERTED => '0', -- Optional inversion for RST_DLY
    IS_RST_INVERTED => '0',     -- Optional inversion for RST
    OUTPUT_PHASE_90 => "FALSE", -- Delays the output phase by 90-degrees
    REFCLK_FREQUENCY => 300.0,  -- Specification of the reference clock frequency in MHz (200.0-2667.0)
    SIM_DEVICE => "ULTRASCALE", -- Set the device version (ULTRASCALE, ULTRASCALE_PLUS_ES1)
    UPDATE_MODE => "ASYNC"      -- Determines when updates to the delay will take effect (ASYNC, MANUAL, SYNC)
)
port map (
    BIT_CTRL_OUT => BIT_CTRL_OUT, -- 40-bit output: Output bus to BITSlice_CONTROL
    CNTVALUEOUT => CNTVALUEOUT,   -- 9-bit output: Counter value to device logic
    TRI_OUT => TRI_OUT,           -- 1-bit output: Output to the TBYTE_IN pins of the bitslices
    BIT_CTRL_IN => BIT_CTRL_IN,   -- 40-bit input: Input bus from BITSlice_CONTROL

```

```

        CE => CE,                -- 1-bit input: Active high enable increment/decrement input
        CLK => CLK,              -- 1-bit input: Clock input
        CNTVALUEIN => CNTVALUEIN, -- 9-bit input: Counter value input
        EN_VTC => EN_VTC,        -- 1-bit input: Enable to keep stable delay over VT
        INC => INC,              -- 1-bit input: Increment the current delay tap setting
        LOAD => LOAD,            -- 1-bit input: Load the CNTVALUEIN tap setting
        RST => RST,              -- 1-bit input: Asynchronous assert, synchronous deassert
        RST_DLY => RST_DLY       -- 1-bit input: Reset the internal DELAY value to DELAY_VALUE
    );

-- End of TX_BITSLICE_TRI_inst instantiation
    
```

## Verilog Instantiation Template

```

// TX_BITSLICE_TRI: TX_BITSLICE_TRI for tristate using Native Mode
//                               UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

TX_BITSLICE_TRI #(
    .DATA_WIDTH(8),                // Parallel data input width (4-8)
    .DELAY_FORMAT("TIME"),         // Units of the DELAY_VALUE (COUNT, TIME)
    .DELAY_TYPE("FIXED"),         // Set the type of tap delay line (FIXED, VARIABLE, VAR_LOAD)
    .DELAY_VALUE(0),              // Output delay value setting
    .INIT(1'b1),                  // Defines initial 0 value
    .IS_CLK_INVERTED(1'b0),       // Optional inversion for CLK
    .IS_RST_DLY_INVERTED(1'b0),   // Optional inversion for RST_DLY
    .IS_RST_INVERTED(1'b0),       // Optional inversion for RST
    .OUTPUT_PHASE_90("FALSE"),    // Delays the output phase by 90-degrees
    .REFCLK_FREQUENCY(300.0),     // Specification of the reference clock frequency in MHz (200.0-2667.0)
    .SIM_DEVICE("ULTRASCALE"),    // Set the device version (ULTRASCALE, ULTRASCALE_PLUS_ES1)
    .UPDATE_MODE("ASYNC")        // Determines when updates to the delay will take effect (ASYNC, MANUAL,
                                // SYNC)
)
TX_BITSLICE_TRI_inst (
    .BIT_CTRL_OUT(BIT_CTRL_OUT), // 40-bit output: Output bus to BITSlice_CONTROL
    .CNTVALUEOUT(CNTVALUEOUT),   // 9-bit output: Counter value to device logic
    .TRI_OUT(TRI_OUT),           // 1-bit output: Output to the TBYTE_IN pins of the bitslices
    .BIT_CTRL_IN(BIT_CTRL_IN),  // 40-bit input: Input bus from BITSlice_CONTROL
    .CE(CE),                     // 1-bit input: Active high enable increment/decrement input
    .CLK(CLK),                   // 1-bit input: Clock input
    .CNTVALUEIN(CNTVALUEIN),     // 9-bit input: Counter value input
    .EN_VTC(EN_VTC),            // 1-bit input: Enable to keep stable delay over VT
    .INC(INC),                   // 1-bit input: Increment the current delay tap setting
    .LOAD(LOAD),                // 1-bit input: Load the CNTVALUEIN tap setting
    .RST(RST),                   // 1-bit input: Asynchronous assert, synchronous deassert
    .RST_DLY(RST_DLY)           // 1-bit input: Reset the internal DELAY value to DELAY_VALUE
);

// End of TX_BITSLICE_TRI_inst instantiation
    
```

## For More Information

- See the *UltraScale Architecture SelectIO Resources User Guide* ([UG571](#)).
- See the [UltraScale Programmable Devices Documentation](#).

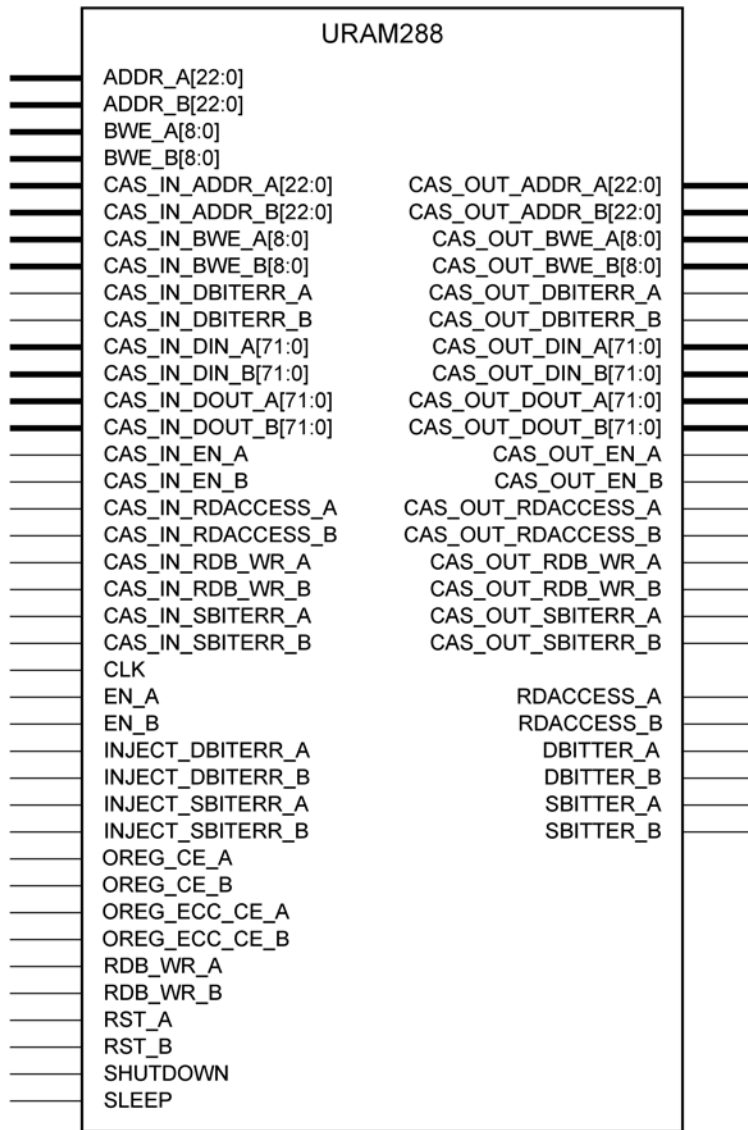
# URAM288

Primitive: 288K-bit High-Density Memory Building Block

PRIMITIVE\_GROUP: BLOCKRAM

PRIMITIVE\_SUBGROUP: URAM

Families: UltraScale+



X15116-102615

## Introduction

288K-bit High-Density Base Memory Building Block

## Port Descriptions

Port	Direction	Width	Function
ADDR_A<22:0>	Input	23	Port A address
ADDR_B<22:0>	Input	23	Port B address
BWE_A<8:0>	Input	9	Port A Byte-write-enable
BWE_B<8:0>	Input	9	Port A Byte-write-enable
CAS_IN_ADDR_A<22:0>	Input	23	Port A cascade input address
CAS_IN_ADDR_B<22:0>	Input	23	Port B cascade input address
CAS_IN_BWE_A<8:0>	Input	9	Port A cascade Byte-write enable input
CAS_IN_BWE_B<8:0>	Input	9	Port B cascade Byte-write enable input
CAS_IN_DBITERR_A	Input	1	Port A cascade double-bit error flag input
CAS_IN_DBITERR_B	Input	1	Port B cascade double-bit error flag input
CAS_IN_DIN_A<71:0>	Input	72	Port A cascade input write mode data
CAS_IN_DIN_B<71:0>	Input	72	Port B cascade input write mode data
CAS_IN_DOUT_A<71:0>	Input	72	Port A cascade input read mode data
CAS_IN_DOUT_B<71:0>	Input	72	Port B cascade input read mode data
CAS_IN_EN_A	Input	1	Port A cascade enable input
CAS_IN_EN_B	Input	1	Port B cascade enable input
CAS_IN_RDACCESS_A	Input	1	Port A cascade read status input
CAS_IN_RDACCESS_B	Input	1	Port B cascade read status input
CAS_IN_RDB_WR_A	Input	1	Port A cascade read/write select input
CAS_IN_RDB_WR_B	Input	1	Port A cascade read/write select input
CAS_IN_SBITERR_A	Input	1	Port A cascade single-bit error flag input
CAS_IN_SBITERR_B	Input	1	Port B cascade single-bit error flag input
CAS_OUT_ADDR_A<22:0>	Output	23	Port A cascade output address
CAS_OUT_ADDR_B<22:0>	Output	23	Port B cascade output address
CAS_OUT_BWE_A<8:0>	Output	9	Port A cascade Byte-write enable output
CAS_OUT_BWE_B<8:0>	Output	9	Port B cascade Byte-write enable output
CAS_OUT_DBITERR_A	Output	1	Port A cascade double-bit error flag output
CAS_OUT_DBITERR_B	Output	1	Port B cascade double-bit error flag output
CAS_OUT_DIN_A<71:0>	Output	72	Port A cascade output write mode data
CAS_OUT_DIN_B<71:0>	Output	72	Port B cascade output write mode data
CAS_OUT_DOUT_A<71:0>	Output	72	Port A cascade output read mode data
CAS_OUT_DOUT_B<71:0>	Output	72	Port B cascade output read mode data
CAS_OUT_EN_A	Output	1	Port A cascade output enable
CAS_OUT_EN_B	Output	1	Port B cascade output enable
CAS_OUT_RDACCESS_A	Output	1	Port A cascade read status output

Port	Direction	Width	Function
CAS_OUT_RDACCESS_B	Output	1	Port B cascade read status output
CAS_OUT_RDB_WR_A	Output	1	Port A cascade read/write select output
CAS_OUT_RDB_WR_B	Output	1	Port B cascade read/write select output
CAS_OUT_SBITERR_A	Output	1	Port A cascade single-bit error flag output
CAS_OUT_SBITERR_B	Output	1	Port B cascade single-bit error flag output
CLK	Input	1	Clock source
DBITERR_A	Output	1	Port A double-bit error flag status
DBITERR_B	Output	1	Port B double-bit error flag status
DIN_A<71:0>	Input	72	Port A write data input
DIN_B<71:0>	Input	72	Port B write data input
DOUT_A<71:0>	Output	72	Port A read data output
DOUT_B<71:0>	Output	72	Port B read data output
EN_A	Input	1	Port A enable
EN_B	Input	1	Port B enable
INJECT_DBITERR_A	Input	1	Port A double-bit error injection
INJECT_DBITERR_B	Input	1	Port B double-bit error injection
INJECT_SBITERR_A	Input	1	Port A single-bit error injection
INJECT_SBITERR_B	Input	1	Port B single-bit error injection
OREG_CE_A	Input	1	Port A output register clock enable
OREG_CE_B	Input	1	Port B output register clock enable
OREG_ECC_CE_A	Input	1	Port A ECC decoder output register clock enable
OREG_ECC_CE_B	Input	1	Port B ECC decoder output register clock enable
RDACCESS_A	Output	1	Port A read status
RDACCESS_B	Output	1	Port B read status
RDB_WR_A	Input	1	Port A read/write select
RDB_WR_B	Input	1	Port B read/write select
RST_A	Input	1	Port A asynchronous or synchronous reset for output registers
RST_B	Input	1	Port B asynchronous or synchronous reset for output registers
SBITERR_A	Output	1	Port A single-bit error flag status
SBITERR_B	Output	1	Port B single-bit error flag status
SLEEP	Input	1	Dynamic power gating control



## Design Entry Method

Instantiation	Yes
Inference	Yes
IP and IP Integrator Catalog	Recommended

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
AUTO_SLEEP_LATENCY	DECIMAL	8, 3, 4, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15	8	Latency requirement to enter sleep mode
AVG_CONS_INACTIVE_CYCLES	DECIMAL	10 to 100000	10	Average consecutive inactive cycles when is SLEEP mode for power estimation
BWE_MODE_A	STRING	"PARITY_INTERLEAVED", "PARITY_INDEPENDENT"	"PARITY_INTERLEAVED"	Port A Byte write control
BWE_MODE_B	STRING	"PARITY_INTERLEAVED", "PARITY_INDEPENDENT"	"PARITY_INTERLEAVED"	Port B Byte write control
CASCADE_ORDER_A	STRING	"NONE", "FIRST", "LAST", "MIDDLE"	"NONE"	Port A position in cascade chain
CASCADE_ORDER_B	STRING	"NONE", "FIRST", "LAST", "MIDDLE"	"NONE"	Port B position in cascade chain
EN_AUTO_SLEEP_MODE	STRING	"FALSE", "TRUE"	"FALSE"	Enable to automatically enter sleep mode
EN_ECC_RD_A	STRING	"FALSE", "TRUE"	"FALSE"	Port A ECC encoder
EN_ECC_RD_B	STRING	"FALSE", "TRUE"	"FALSE"	Port B ECC encoder
EN_ECC_WR_A	STRING	"FALSE", "TRUE"	"FALSE"	Port A ECC decoder
EN_ECC_WR_B	STRING	"FALSE", "TRUE"	"FALSE"	Port B ECC decoder
IREG_PRE_A	STRING	"FALSE", "TRUE"	"FALSE"	Optional Port A input pipeline registers
IREG_PRE_B	STRING	"FALSE", "TRUE"	"FALSE"	Optional Port B input pipeline registers
IS_CLK_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Optional inverter for CLK
IS_EN_A_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Optional inverter for Port A enable
IS_EN_B_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Optional inverter for Port B enable
IS_RDB_WR_A_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Optional inverter for Port A read/write select
IS_RDB_WR_B_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Optional inverter for Port B read/write select

Attribute	Type	Allowed Values	Default	Description
IS_RST_A_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Optional inverter for Port A reset
IS_RST_B_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Optional inverter for Port B reset
MATRIX_ID	STRING	String	"NONE"	!UNKNOWN!
NUM_UNIQUE_SELF_ADDR_A	DECIMAL	1 to 2048	1	!UNKNOWN!
NUM_UNIQUE_SELF_ADDR_B	DECIMAL	1 to 2048	1	!UNKNOWN!
NUM_URAM_IN_MATRIX	DECIMAL	1 to 2048	1	!UNKNOWN!
OREG_A	STRING	"FALSE", "TRUE"	"FALSE"	Optional Port A output pipeline registers
OREG_B	STRING	"FALSE", "TRUE"	"FALSE"	Optional Port B output pipeline registers
OREG_ECC_A	STRING	"FALSE", "TRUE"	"FALSE"	Port A ECC decoder output
OREG_ECC_B	STRING	"FALSE", "TRUE"	"FALSE"	Port B output ECC decoder
REG_CAS_A	STRING	"FALSE", "TRUE"	"FALSE"	Optional Port A cascade register
REG_CAS_B	STRING	"FALSE", "TRUE"	"FALSE"	Optional Port B cascade register
RST_MODE_A	STRING	"SYNC", "ASYNC"	"SYNC"	Port A reset mode
RST_MODE_B	STRING	"SYNC", "ASYNC"	"SYNC"	Port B reset mode
SELF_ADDR_A	HEX	11'h000 to 11'h7ff	11'h000	Port A self-address value
SELF_ADDR_B	HEX	11'h000 to 11'h7ff	11'h000	Port B self-address value
SELF_MASK_A	HEX	11'h000 to 11'h7ff	11'h7ff	Port A self-address mask
SELF_MASK_B	HEX	11'h000 to 11'h7ff	11'h7ff	Port B self-address mask
USE_EXT_CE_A	STRING	"FALSE", "TRUE"	"FALSE"	Enable Port A external CE inputs for output registers
USE_EXT_CE_B	STRING	"FALSE", "TRUE"	"FALSE"	Enable Port B external CE inputs for output registersE

## VHDL Instantiation Template

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- URAM288: 288K-bit High-Density Memory Building Block
-- UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1
    
```

```

URAM288_inst : URAM288
generic map (
    AUTO_SLEEP_LATENCY => 8,           -- Latency requirement to enter sleep mode
    AVG_CONS_INACTIVE_CYCLES => 10,    -- Average consecutive inactive cycles when in SLEEP mode for power
                                        -- estimation
    BWE_MODE_A => "PARITY_INTERLEAVED", -- Port A Byte write control
    BWE_MODE_B => "PARITY_INTERLEAVED", -- Port B Byte write control
    CASCADE_ORDER_A => "NONE",         -- Port A position in cascade chain
    CASCADE_ORDER_B => "NONE",         -- Port B position in cascade chain
    EN_AUTO_SLEEP_MODE => "FALSE",     -- Enable to automatically enter sleep mode
    EN_ECC_RD_A => "FALSE",            -- Port A ECC encoder
    EN_ECC_RD_B => "FALSE",            -- Port B ECC encoder
    EN_ECC_WR_A => "FALSE",            -- Port A ECC decoder
    EN_ECC_WR_B => "FALSE",            -- Port B ECC decoder
    IREG_PRE_A => "FALSE",             -- Optional Port A input pipeline registers
    IREG_PRE_B => "FALSE",             -- Optional Port B input pipeline registers
    IS_CLK_INVERTED => '0',           -- Optional inverter for CLK
    IS_EN_A_INVERTED => '0',          -- Optional inverter for Port A enable
    IS_EN_B_INVERTED => '0',          -- Optional inverter for Port B enable
    IS_RDB_WR_A_INVERTED => '0',      -- Optional inverter for Port A read/write select
    IS_RDB_WR_B_INVERTED => '0',      -- Optional inverter for Port B read/write select
    IS_RST_A_INVERTED => '0',         -- Optional inverter for Port A reset
    IS_RST_B_INVERTED => '0',         -- Optional inverter for Port B reset
    MATRIX_ID => "NONE",
    NUM_UNIQUE_SELF_ADDR_A => 1,
    NUM_UNIQUE_SELF_ADDR_B => 1,
    NUM_URAM_IN_MATRIX => 1,
    OREG_A => "FALSE",                -- Optional Port A output pipeline registers
    OREG_B => "FALSE",                -- Optional Port B output pipeline registers
    OREG_ECC_A => "FALSE",            -- Port A ECC decoder output
    OREG_ECC_B => "FALSE",            -- Port B output ECC decoder
    REG_CAS_A => "FALSE",             -- Optional Port A cascade register
    REG_CAS_B => "FALSE",             -- Optional Port B cascade register
    RST_MODE_A => "SYNC",             -- Port A reset mode
    RST_MODE_B => "SYNC",             -- Port B reset mode
    SELF_ADDR_A => X"000",            -- Port A self-address value
    SELF_ADDR_B => X"000",            -- Port B self-address value
    SELF_MASK_A => X"7ff",           -- Port A self-address mask
    SELF_MASK_B => X"7ff",           -- Port B self-address mask
    USE_EXT_CE_A => "FALSE",          -- Enable Port A external CE inputs for output registers
    USE_EXT_CE_B => "FALSE",          -- Enable Port B external CE inputs for output registers
)
port map (
    CAS_OUT_ADDR_A => CAS_OUT_ADDR_A, -- 23-bit output: Port A cascade output address
    CAS_OUT_ADDR_B => CAS_OUT_ADDR_B, -- 23-bit output: Port B cascade output address
    CAS_OUT_BWE_A => CAS_OUT_BWE_A,   -- 9-bit output: Port A cascade Byte-write enable output
    CAS_OUT_BWE_B => CAS_OUT_BWE_B,   -- 9-bit output: Port B cascade Byte-write enable output
    CAS_OUT_DBITERR_A => CAS_OUT_DBITERR_A, -- 1-bit output: Port A cascade double-bit error flag output
    CAS_OUT_DBITERR_B => CAS_OUT_DBITERR_B, -- 1-bit output: Port B cascade double-bit error flag output
    CAS_OUT_DIN_A => CAS_OUT_DIN_A,   -- 72-bit output: Port A cascade output write mode data
    CAS_OUT_DIN_B => CAS_OUT_DIN_B,   -- 72-bit output: Port B cascade output write mode data
    CAS_OUT_DOUT_A => CAS_OUT_DOUT_A, -- 72-bit output: Port A cascade output read mode data
    CAS_OUT_DOUT_B => CAS_OUT_DOUT_B, -- 72-bit output: Port B cascade output read mode data
    CAS_OUT_EN_A => CAS_OUT_EN_A,     -- 1-bit output: Port A cascade output enable
    CAS_OUT_EN_B => CAS_OUT_EN_B,     -- 1-bit output: Port B cascade output enable
    CAS_OUT_RDACCESS_A => CAS_OUT_RDACCESS_A, -- 1-bit output: Port A cascade read status output
    CAS_OUT_RDACCESS_B => CAS_OUT_RDACCESS_B, -- 1-bit output: Port B cascade read status output
    CAS_OUT_RDB_WR_A => CAS_OUT_RDB_WR_A, -- 1-bit output: Port A cascade read/write select output
    CAS_OUT_RDB_WR_B => CAS_OUT_RDB_WR_B, -- 1-bit output: Port B cascade read/write select output
    CAS_OUT_SBITERR_A => CAS_OUT_SBITERR_A, -- 1-bit output: Port A cascade single-bit error flag output
    CAS_OUT_SBITERR_B => CAS_OUT_SBITERR_B, -- 1-bit output: Port B cascade single-bit error flag output
    DBITERR_A => DBITERR_A,          -- 1-bit output: Port A double-bit error flag status
    DBITERR_B => DBITERR_B,          -- 1-bit output: Port B double-bit error flag status
    DOUT_A => DOUT_A,                -- 72-bit output: Port A read data output
    DOUT_B => DOUT_B,                -- 72-bit output: Port B read data output
    RDACCESS_A => RDACCESS_A,        -- 1-bit output: Port A read status
    RDACCESS_B => RDACCESS_B,        -- 1-bit output: Port B read status
    SBITERR_A => SBITERR_A,          -- 1-bit output: Port A single-bit error flag status
    SBITERR_B => SBITERR_B,          -- 1-bit output: Port B single-bit error flag status
    ADDR_A => ADDR_A,                -- 23-bit input: Port A address
    ADDR_B => ADDR_B,                -- 23-bit input: Port B address
    BWE_A => BWE_A,                  -- 9-bit input: Port A Byte-write enable

```

```

BWE_B => BWE_B,
CAS_IN_ADDR_A => CAS_IN_ADDR_A,
CAS_IN_ADDR_B => CAS_IN_ADDR_B,
CAS_IN_BWE_A => CAS_IN_BWE_A,
CAS_IN_BWE_B => CAS_IN_BWE_B,
CAS_IN_DBITERR_A => CAS_IN_DBITERR_A,
CAS_IN_DBITERR_B => CAS_IN_DBITERR_B,
CAS_IN_DIN_A => CAS_IN_DIN_A,
CAS_IN_DIN_B => CAS_IN_DIN_B,
CAS_IN_DOUT_A => CAS_IN_DOUT_A,
CAS_IN_DOUT_B => CAS_IN_DOUT_B,
CAS_IN_EN_A => CAS_IN_EN_A,
CAS_IN_EN_B => CAS_IN_EN_B,
CAS_IN_RDACCESS_A => CAS_IN_RDACCESS_A,
CAS_IN_RDACCESS_B => CAS_IN_RDACCESS_B,
CAS_IN_RDB_WR_A => CAS_IN_RDB_WR_A,
CAS_IN_RDB_WR_B => CAS_IN_RDB_WR_B,
CAS_IN_SBITERR_A => CAS_IN_SBITERR_A,
CAS_IN_SBITERR_B => CAS_IN_SBITERR_B,
CLK => CLK,
DIN_A => DIN_A,
DIN_B => DIN_B,
EN_A => EN_A,
EN_B => EN_B,
INJECT_DBITERR_A => INJECT_DBITERR_A,
INJECT_DBITERR_B => INJECT_DBITERR_B,
INJECT_SBITERR_A => INJECT_SBITERR_A,
INJECT_SBITERR_B => INJECT_SBITERR_B,
OREG_CE_A => OREG_CE_A,
OREG_CE_B => OREG_CE_B,
OREG_ECC_CE_A => OREG_ECC_CE_A,
OREG_ECC_CE_B => OREG_ECC_CE_B,
RDB_WR_A => RDB_WR_A,
RDB_WR_B => RDB_WR_B,
RST_A => RST_A,

RST_B => RST_B,

SLEEP => SLEEP
);

-- End of URAM288_inst instantiation
    
```

## Verilog Instantiation Template

```

// URAM288: 288K-bit High-Density Memory Building Block
// UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

URAM288 #(
    .AUTO_SLEEP_LATENCY(8), // Latency requirement to enter sleep mode
    .AVG_CONS_INACTIVE_CYCLES(10), // Average consecutive inactive cycles when is SLEEP mode for power
    // estimation
    .BWE_MODE_A("PARITY_INTERLEAVED"), // Port A Byte write control
    .BWE_MODE_B("PARITY_INTERLEAVED"), // Port B Byte write control
    .CASCADE_ORDER_A("NONE"), // Port A position in cascade chain
    .CASCADE_ORDER_B("NONE"), // Port B position in cascade chain
    .EN_AUTO_SLEEP_MODE("FALSE"), // Enable to automatically enter sleep mode
    .EN_ECC_RD_A("FALSE"), // Port A ECC encoder
    .EN_ECC_RD_B("FALSE"), // Port B ECC encoder
    .EN_ECC_WR_A("FALSE"), // Port A ECC decoder
    .EN_ECC_WR_B("FALSE"), // Port B ECC decoder
    .IREG_PRE_A("FALSE"), // Optional Port A input pipeline registers
    .IREG_PRE_B("FALSE"), // Optional Port B input pipeline registers
    .IS_CLK_INVERTED(1'b0), // Optional inverter for CLK
    .IS_EN_A_INVERTED(1'b0), // Optional inverter for Port A enable
    .IS_EN_B_INVERTED(1'b0), // Optional inverter for Port B enable
    .IS_RDB_WR_A_INVERTED(1'b0), // Optional inverter for Port A read/write select
    
```

```

        .IS_RDB_WR_B_INVERTED(1'b0), // Optional inverter for Port B read/write select
        .IS_RST_A_INVERTED(1'b0), // Optional inverter for Port A reset
        .IS_RST_B_INVERTED(1'b0), // Optional inverter for Port B reset
        .MATRIX_ID("NONE"),
        .NUM_UNIQUE_SELF_ADDR_A(1),
        .NUM_UNIQUE_SELF_ADDR_B(1),
        .NUM_URAM_IN_MATRIX(1),
        .OREG_A("FALSE"), // Optional Port A output pipeline registers
        .OREG_B("FALSE"), // Optional Port B output pipeline registers
        .OREG_ECC_A("FALSE"), // Port A ECC decoder output
        .OREG_ECC_B("FALSE"), // Port B output ECC decoder
        .REG_CAS_A("FALSE"), // Optional Port A cascade register
        .REG_CAS_B("FALSE"), // Optional Port B cascade register
        .RST_MODE_A("SYNC"), // Port A reset mode
        .RST_MODE_B("SYNC"), // Port B reset mode
        .SELF_ADDR_A(11'h000), // Port A self-address value
        .SELF_ADDR_B(11'h000), // Port B self-address value
        .SELF_MASK_A(11'h7ff), // Port A self-address mask
        .SELF_MASK_B(11'h7ff), // Port B self-address mask
        .USE_EXT_CE_A("FALSE"), // Enable Port A external CE inputs for output registers
        .USE_EXT_CE_B("FALSE") // Enable Port B external CE inputs for output registers
    )
    URAM288_inst (
        .CAS_OUT_ADDR_A(CAS_OUT_ADDR_A), // 23-bit output: Port A cascade output address
        .CAS_OUT_ADDR_B(CAS_OUT_ADDR_B), // 23-bit output: Port B cascade output address
        .CAS_OUT_BWE_A(CAS_OUT_BWE_A), // 9-bit output: Port A cascade Byte-write enable output
        .CAS_OUT_BWE_B(CAS_OUT_BWE_B), // 9-bit output: Port B cascade Byte-write enable output
        .CAS_OUT_DBITERR_A(CAS_OUT_DBITERR_A), // 1-bit output: Port A cascade double-bit error flag output
        .CAS_OUT_DBITERR_B(CAS_OUT_DBITERR_B), // 1-bit output: Port B cascade double-bit error flag output
        .CAS_OUT_DIN_A(CAS_OUT_DIN_A), // 72-bit output: Port A cascade output write mode data
        .CAS_OUT_DIN_B(CAS_OUT_DIN_B), // 72-bit output: Port B cascade output write mode data
        .CAS_OUT_DOUT_A(CAS_OUT_DOUT_A), // 72-bit output: Port A cascade output read mode data
        .CAS_OUT_DOUT_B(CAS_OUT_DOUT_B), // 72-bit output: Port B cascade output read mode data
        .CAS_OUT_EN_A(CAS_OUT_EN_A), // 1-bit output: Port A cascade output enable
        .CAS_OUT_EN_B(CAS_OUT_EN_B), // 1-bit output: Port B cascade output enable
        .CAS_OUT_RDACCESS_A(CAS_OUT_RDACCESS_A), // 1-bit output: Port A cascade read status output
        .CAS_OUT_RDACCESS_B(CAS_OUT_RDACCESS_B), // 1-bit output: Port B cascade read status output
        .CAS_OUT_RDB_WR_A(CAS_OUT_RDB_WR_A), // 1-bit output: Port A cascade read/write select output
        .CAS_OUT_RDB_WR_B(CAS_OUT_RDB_WR_B), // 1-bit output: Port B cascade read/write select output
        .CAS_OUT_SBITERR_A(CAS_OUT_SBITERR_A), // 1-bit output: Port A cascade single-bit error flag output
        .CAS_OUT_SBITERR_B(CAS_OUT_SBITERR_B), // 1-bit output: Port B cascade single-bit error flag output
        .DBITERR_A(DBITERR_A), // 1-bit output: Port A double-bit error flag status
        .DBITERR_B(DBITERR_B), // 1-bit output: Port B double-bit error flag status
        .DOUT_A(DOUT_A), // 72-bit output: Port A read data output
        .DOUT_B(DOUT_B), // 72-bit output: Port B read data output
        .RDACCESS_A(RDACCESS_A), // 1-bit output: Port A read status
        .RDACCESS_B(RDACCESS_B), // 1-bit output: Port B read status
        .SBITERR_A(SBITERR_A), // 1-bit output: Port A single-bit error flag status
        .SBITERR_B(SBITERR_B), // 1-bit output: Port B single-bit error flag status
        .ADDR_A(ADDR_A), // 23-bit input: Port A address
        .ADDR_B(ADDR_B), // 23-bit input: Port B address
        .BWE_A(BWE_A), // 9-bit input: Port A Byte-write enable
        .BWE_B(BWE_B), // 9-bit input: Port B Byte-write enable
        .CAS_IN_ADDR_A(CAS_IN_ADDR_A), // 23-bit input: Port A cascade input address
        .CAS_IN_ADDR_B(CAS_IN_ADDR_B), // 23-bit input: Port B cascade input address
        .CAS_IN_BWE_A(CAS_IN_BWE_A), // 9-bit input: Port A cascade Byte-write enable input
        .CAS_IN_BWE_B(CAS_IN_BWE_B), // 9-bit input: Port B cascade Byte-write enable input
        .CAS_IN_DBITERR_A(CAS_IN_DBITERR_A), // 1-bit input: Port A cascade double-bit error flag input
        .CAS_IN_DBITERR_B(CAS_IN_DBITERR_B), // 1-bit input: Port B cascade double-bit error flag input
        .CAS_IN_DIN_A(CAS_IN_DIN_A), // 72-bit input: Port A cascade input write mode data
        .CAS_IN_DIN_B(CAS_IN_DIN_B), // 72-bit input: Port B cascade input write mode data
        .CAS_IN_DOUT_A(CAS_IN_DOUT_A), // 72-bit input: Port A cascade input read mode data
        .CAS_IN_DOUT_B(CAS_IN_DOUT_B), // 72-bit input: Port B cascade input read mode data
        .CAS_IN_EN_A(CAS_IN_EN_A), // 1-bit input: Port A cascade enable input
        .CAS_IN_EN_B(CAS_IN_EN_B), // 1-bit input: Port B cascade enable input
        .CAS_IN_RDACCESS_A(CAS_IN_RDACCESS_A), // 1-bit input: Port A cascade read status input
        .CAS_IN_RDACCESS_B(CAS_IN_RDACCESS_B), // 1-bit input: Port B cascade read status input
        .CAS_IN_RDB_WR_A(CAS_IN_RDB_WR_A), // 1-bit input: Port A cascade read/write select input
        .CAS_IN_RDB_WR_B(CAS_IN_RDB_WR_B), // 1-bit input: Port A cascade read/write select input
        .CAS_IN_SBITERR_A(CAS_IN_SBITERR_A), // 1-bit input: Port A cascade single-bit error flag input
        .CAS_IN_SBITERR_B(CAS_IN_SBITERR_B), // 1-bit input: Port B cascade single-bit error flag input
        .CLK(CLK), // 1-bit input: Clock source
    )

```

```

.DIN_A(DIN_A), // 72-bit input: Port A write data input
.DIN_B(DIN_B), // 72-bit input: Port B write data input
.EN_A(EN_A), // 1-bit input: Port A enable
.EN_B(EN_B), // 1-bit input: Port B enable
.INJECT_DBITERR_A(INJECT_DBITERR_A), // 1-bit input: Port A double-bit error injection
.INJECT_DBITERR_B(INJECT_DBITERR_B), // 1-bit input: Port B double-bit error injection
.INJECT_SBITERR_A(INJECT_SBITERR_A), // 1-bit input: Port A single-bit error injection
.INJECT_SBITERR_B(INJECT_SBITERR_B), // 1-bit input: Port B single-bit error injection
.OREG_CE_A(OREG_CE_A), // 1-bit input: Port A output register clock enable
.OREG_CE_B(OREG_CE_B), // 1-bit input: Port B output register clock enable
.OREG_ECC_CE_A(OREG_ECC_CE_A), // 1-bit input: Port A ECC decoder output register clock enable
.OREG_ECC_CE_B(OREG_ECC_CE_B), // 1-bit input: Port B ECC decoder output register clock enable
.RDB_WR_A(RDB_WR_A), // 1-bit input: Port A read/write select
.RDB_WR_B(RDB_WR_B), // 1-bit input: Port B read/write select
.RST_A(RST_A), // 1-bit input: Port A asynchronous or synchronous reset for
// output registers

.RST_B(RST_B), // 1-bit input: Port B asynchronous or synchronous reset for
// output registers

.SLEEP(SLEEP) // 1-bit input: Dynamic power gating control
);

// End of URAM288_inst instantiation
    
```

## For More Information

- See the *UltraScale Architecture Memory Resources User Guide* ([UG573](#)).
- See the [UltraScale Programmable Devices Documentation](#).

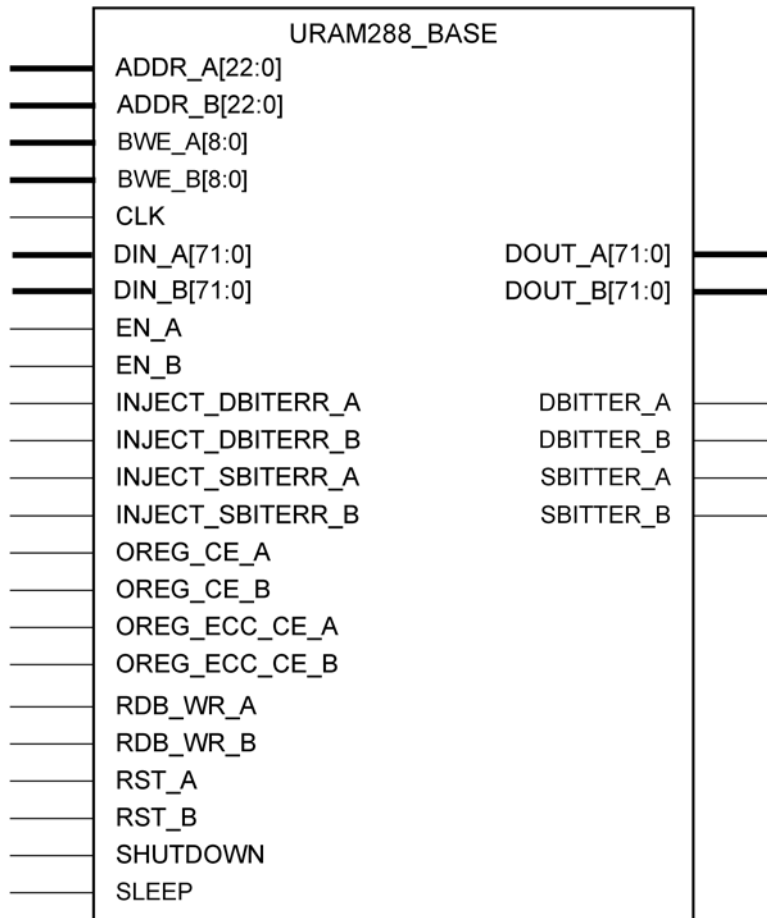
# URAM288\_BASE

Primitive: 288K-bit High-Density Base Memory Building Block

PRIMITIVE\_GROUP: BLOCKRAM

PRIMITIVE\_SUBGROUP: URAM

Families: UltraScale+



X15117-102815

## Introduction

288K-bit High-Density Base Memory Building Block

## Port Descriptions

Port	Direction	Width	Function
ADDR_A<22:0>	Input	23	Port A address
ADDR_B<22:0>	Input	23	Port B address
BWE_A<8:0>	Input	9	Port A Byte-write enable
BWE_B<8:0>	Input	9	Port B Byte-write enable
CLK	Input	1	Clock source
DBITERR_A	Output	1	Port A double-bit error flag status
DBITERR_B	Output	1	Port B double-bit error flag status
DIN_A<71:0>	Input	72	Port A write data input
DIN_B<71:0>	Input	72	Port B write data input
DOUT_A<71:0>	Output	72	Port A read data output
DOUT_B<71:0>	Output	72	Port B read data output
EN_A	Input	1	Port A enable
EN_B	Input	1	Port B enable
INJECT_DBITERR_A	Input	1	Port A double-bit error injection
INJECT_DBITERR_B	Input	1	Port B double-bit error injection
INJECT_SBITERR_A	Input	1	Port A single-bit error injection
INJECT_SBITERR_B	Input	1	Port B single-bit error injection
OREG_CE_A	Input	1	Port A output register clock enable
OREG_CE_B	Input	1	Port B output register clock enable
OREG_ECC_CE_A	Input	1	Port A ECC decoder output register clock enable
OREG_ECC_CE_B	Input	1	Port B ECC decoder output register clock enable
RDB_WR_A	Input	1	Port A read/write select
RDB_WR_B	Input	1	Port B read/write select
RST_A	Input	1	Port A asynchronous or synchronous reset for output registers
RST_B	Input	1	Port B asynchronous or synchronous reset for output registers
SBITERR_A	Output	1	Port A single-bit error flag status
SBITERR_B	Output	1	Port B single-bit error flag status
SLEEP	Input	1	Dynamic power gating control

## Design Entry Method

Instantiation	Yes
Inference	Yes
IP and IP Integrator Catalog	Recommended



## Available Attributes

Attribute	Type	Allowed Values	Default	Description
AUTO_SLEEP_LATENCY	DECIMAL	8, 3, 4, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15	8	Latency requirement to enter sleep mode
AVG_CONS_INACTIVE_CYCLES	DECIMAL	10 to 100000	10	Average consecutive inactive cycles when in SLEEP mode for power estimation
BWE_MODE_A	STRING	"PARITY_INTERLEAVED", "PARITY_INDEPENDENT"	"PARITY_INTERLEAVED"	Port A Byte write control
BWE_MODE_B	STRING	"PARITY_INTERLEAVED", "PARITY_INDEPENDENT"	"PARITY_INTERLEAVED"	Port B Byte write control
EN_AUTO_SLEEP_MODE	STRING	"FALSE", "TRUE"	"FALSE"	Enable to automatically enter sleep mode
EN_ECC_RD_A	STRING	"FALSE", "TRUE"	"FALSE"	Port A ECC encoder
EN_ECC_RD_B	STRING	"FALSE", "TRUE"	"FALSE"	Port B ECC encoder
EN_ECC_WR_A	STRING	"FALSE", "TRUE"	"FALSE"	Port A ECC decoder
EN_ECC_WR_B	STRING	"FALSE", "TRUE"	"FALSE"	Port B ECC decoder
IREG_PRE_A	STRING	"FALSE", "TRUE"	"FALSE"	Optional Port A input pipeline registers
IREG_PRE_B	STRING	"FALSE", "TRUE"	"FALSE"	Optional Port B input pipeline registers
IS_CLK_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Optional inverter for CLK
IS_EN_A_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Optional inverter for Port A enable
IS_EN_B_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Optional inverter for Port B enable
IS_RDB_WR_A_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Optional inverter for Port A read/write select
IS_RDB_WR_B_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Optional inverter for Port B read/write select
IS_RST_A_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Optional inverter for Port A reset
IS_RST_B_INVERTED	BINARY	1'b0 to 1'b1	1'b0	Optional inverter for Port B reset
OREG_A	STRING	"FALSE", "TRUE"	"FALSE"	Optional Port A output pipeline registers
OREG_B	STRING	"FALSE", "TRUE"	"FALSE"	Optional Port B output pipeline registers
OREG_ECC_A	STRING	"FALSE", "TRUE"	"FALSE"	Port A ECC decoder output
OREG_ECC_B	STRING	"FALSE", "TRUE"	"FALSE"	Port B output ECC decoder
RST_MODE_A	STRING	"SYNC", "ASYNC"	"SYNC"	Port A reset mode

Attribute	Type	Allowed Values	Default	Description
RST_MODE_B	STRING	"SYNC", "ASYNC"	"SYNC"	Port B reset mode
USE_EXT_CE_A	STRING	"FALSE", "TRUE"	"FALSE"	Enable Port A external CE inputs for output registers
USE_EXT_CE_B	STRING	"FALSE", "TRUE"	"FALSE"	Enable Port B external CE inputs for output registers

## VHDL Instantiation Template

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- URAM288_BASE: 288K-bit High-Density Base Memory Building Block
-- UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

URAM288_BASE_inst : URAM288_BASE
generic map (
    AUTO_SLEEP_LATENCY => 8,                -- Latency requirement to enter sleep mode
    AVG_CONS_INACTIVE_CYCLES => 10,        -- Average consecutive inactive cycles when is SLEEP mode for power
                                           -- estimation
    BWE_MODE_A => "PARITY_INTERLEAVED",    -- Port A Byte write control
    BWE_MODE_B => "PARITY_INTERLEAVED",    -- Port B Byte write control
    EN_AUTO_SLEEP_MODE => "FALSE",        -- Enable to automatically enter sleep mode
    EN_ECC_RD_A => "FALSE",               -- Port A ECC encoder
    EN_ECC_RD_B => "FALSE",               -- Port B ECC encoder
    EN_ECC_WR_A => "FALSE",               -- Port A ECC decoder
    EN_ECC_WR_B => "FALSE",               -- Port B ECC decoder
    IREG_PRE_A => "FALSE",                 -- Optional Port A input pipeline registers
    IREG_PRE_B => "FALSE",                 -- Optional Port B input pipeline registers
    IS_CLK_INVERTED => '0',                -- Optional inverter for CLK
    IS_EN_A_INVERTED => '0',               -- Optional inverter for Port A enable
    IS_EN_B_INVERTED => '0',               -- Optional inverter for Port B enable
    IS_RDB_WR_A_INVERTED => '0',          -- Optional inverter for Port A read/write select
    IS_RDB_WR_B_INVERTED => '0',          -- Optional inverter for Port B read/write select
    IS_RST_A_INVERTED => '0',              -- Optional inverter for Port A reset
    IS_RST_B_INVERTED => '0',              -- Optional inverter for Port B reset
    MATRIX_ID => "NONE",
    NUM_UNIQUE_SELF_ADDR_A => 1,
    NUM_UNIQUE_SELF_ADDR_B => 1,
    NUM_URAM_IN_MATRIX => 1,
    OREG_A => "FALSE",                     -- Optional Port A output pipeline registers
    OREG_B => "FALSE",                     -- Optional Port B output pipeline registers
    OREG_ECC_A => "FALSE",                 -- Port A ECC decoder output
    OREG_ECC_B => "FALSE",                 -- Port B output ECC decoder
    RST_MODE_A => "SYNC",                  -- Port A reset mode
    RST_MODE_B => "SYNC",                  -- Port B reset mode
    USE_EXT_CE_A => "FALSE",               -- Enable Port A external CE inputs for output registers
    USE_EXT_CE_B => "FALSE"                -- Enable Port B external CE inputs for output registers
)
port map (
    DBITERR_A => DBITERR_A,                -- 1-bit output: Port A double-bit error flag status
    DBITERR_B => DBITERR_B,                -- 1-bit output: Port B double-bit error flag status
    DOUT_A => DOUT_A,                       -- 72-bit output: Port A read data output
    DOUT_B => DOUT_B,                       -- 72-bit output: Port B read data output
    SBITERR_A => SBITERR_A,                 -- 1-bit output: Port A single-bit error flag status
    SBITERR_B => SBITERR_B,                 -- 1-bit output: Port B single-bit error flag status
    ADDR_A => ADDR_A,                       -- 23-bit input: Port A address
    ADDR_B => ADDR_B,                       -- 23-bit input: Port B address
    BWE_A => BWE_A,                         -- 9-bit input: Port A Byte-write enable
    BWE_B => BWE_B,                         -- 9-bit input: Port B Byte-write enable

```

```

CLK => CLK,                -- 1-bit input: Clock source
DIN_A => DIN_A,            -- 72-bit input: Port A write data input
DIN_B => DIN_B,            -- 72-bit input: Port B write data input
EN_A => EN_A,              -- 1-bit input: Port A enable
EN_B => EN_B,              -- 1-bit input: Port B enable
INJECT_DBITERR_A => INJECT_DBITERR_A, -- 1-bit input: Port A double-bit error injection
INJECT_DBITERR_B => INJECT_DBITERR_B, -- 1-bit input: Port B double-bit error injection
INJECT_SBITERR_A => INJECT_SBITERR_A, -- 1-bit input: Port A single-bit error injection
INJECT_SBITERR_B => INJECT_SBITERR_B, -- 1-bit input: Port B single-bit error injection
OREG_CE_A => OREG_CE_A,    -- 1-bit input: Port A output register clock enable
OREG_CE_B => OREG_CE_B,    -- 1-bit input: Port B output register clock enable
OREG_ECC_CE_A => OREG_ECC_CE_A, -- 1-bit input: Port A ECC decoder output register clock enable
OREG_ECC_CE_B => OREG_ECC_CE_B, -- 1-bit input: Port B ECC decoder output register clock enable
RDB_WR_A => RDB_WR_A,      -- 1-bit input: Port A read/write select
RDB_WR_B => RDB_WR_B,      -- 1-bit input: Port B read/write select
RST_A => RST_A,            -- 1-bit input: Port A asynchronous or synchronous reset for output
                                -- registers

RST_B => RST_B,            -- 1-bit input: Port B asynchronous or synchronous reset for output
                                -- registers

SLEEP => SLEEP             -- 1-bit input: Dynamic power gating control
);

-- End of URAM288_BASE_inst instantiation
    
```

## Verilog Instantiation Template

```

// URAM288_BASE: 288K-bit High-Density Base Memory Building Block
// UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

URAM288_BASE #(
    .AUTO_SLEEP_LATENCY(8),          // Latency requirement to enter sleep mode
    .AVG_CONS_INACTIVE_CYCLES(10),   // Average consecutive inactive cycles when is SLEEP mode for power
                                        // estimation
    .BWE_MODE_A("PARITY_INTERLEAVED"), // Port A Byte write control
    .BWE_MODE_B("PARITY_INTERLEAVED"), // Port B Byte write control
    .EN_AUTO_SLEEP_MODE("FALSE"),    // Enable to automatically enter sleep mode
    .EN_ECC_RD_A("FALSE"),           // Port A ECC encoder
    .EN_ECC_RD_B("FALSE"),           // Port B ECC encoder
    .EN_ECC_WR_A("FALSE"),           // Port A ECC decoder
    .EN_ECC_WR_B("FALSE"),           // Port B ECC decoder
    .IREG_PRE_A("FALSE"),            // Optional Port A input pipeline registers
    .IREG_PRE_B("FALSE"),            // Optional Port B input pipeline registers
    .IS_CLK_INVERTED(1'b0),          // Optional inverter for CLK
    .IS_EN_A_INVERTED(1'b0),         // Optional inverter for Port A enable
    .IS_EN_B_INVERTED(1'b0),         // Optional inverter for Port B enable
    .IS_RDB_WR_A_INVERTED(1'b0),     // Optional inverter for Port A read/write select
    .IS_RDB_WR_B_INVERTED(1'b0),     // Optional inverter for Port B read/write select
    .IS_RST_A_INVERTED(1'b0),        // Optional inverter for Port A reset
    .IS_RST_B_INVERTED(1'b0),        // Optional inverter for Port B reset
    .MATRIX_ID("NONE"),
    .NUM_UNIQUE_SELF_ADDR_A(1),
    .NUM_UNIQUE_SELF_ADDR_B(1),
    .NUM_URAM_IN_MATRIX(1),
    .OREG_A("FALSE"),                // Optional Port A output pipeline registers
    .OREG_B("FALSE"),                // Optional Port B output pipeline registers
    .OREG_ECC_A("FALSE"),            // Port A ECC decoder output
    .OREG_ECC_B("FALSE"),            // Port B output ECC decoder
    .RST_MODE_A("SYNC"),             // Port A reset mode
    .RST_MODE_B("SYNC"),             // Port B reset mode
    .USE_EXT_CE_A("FALSE"),          // Enable Port A external CE inputs for output registers
    .USE_EXT_CE_B("FALSE")          // Enable Port B external CE inputs for output registers
)
URAM288_BASE_inst (
    .DBITERR_A(DBITERR_A),           // 1-bit output: Port A double-bit error flag status
    .DBITERR_B(DBITERR_B),           // 1-bit output: Port B double-bit error flag status
    .DOUT_A(DOUT_A),                 // 72-bit output: Port A read data output
    .DOUT_B(DOUT_B),                 // 72-bit output: Port B read data output
    
```

```

.SBITERR_A(SBITERR_A), // 1-bit output: Port A single-bit error flag status
.SBITERR_B(SBITERR_B), // 1-bit output: Port B single-bit error flag status
.ADDR_A(ADDR_A), // 23-bit input: Port A address
.ADDR_B(ADDR_B), // 23-bit input: Port B address
.BWE_A(BWE_A), // 9-bit input: Port A Byte-write enable
.BWE_B(BWE_B), // 9-bit input: Port B Byte-write enable
.CLK(CLK), // 1-bit input: Clock source
.DIN_A(DIN_A), // 72-bit input: Port A write data input
.DIN_B(DIN_B), // 72-bit input: Port B write data input
.EN_A(EN_A), // 1-bit input: Port A enable
.EN_B(EN_B), // 1-bit input: Port B enable
.INJECT_DBITERR_A(INJECT_DBITERR_A), // 1-bit input: Port A double-bit error injection
.INJECT_DBITERR_B(INJECT_DBITERR_B), // 1-bit input: Port B double-bit error injection
.INJECT_SBITERR_A(INJECT_SBITERR_A), // 1-bit input: Port A single-bit error injection
.INJECT_SBITERR_B(INJECT_SBITERR_B), // 1-bit input: Port B single-bit error injection
.OREG_CE_A(OREG_CE_A), // 1-bit input: Port A output register clock enable
.OREG_CE_B(OREG_CE_B), // 1-bit input: Port B output register clock enable
.OREG_ECC_CE_A(OREG_ECC_CE_A), // 1-bit input: Port A ECC decoder output register clock enable
.OREG_ECC_CE_B(OREG_ECC_CE_B), // 1-bit input: Port B ECC decoder output register clock enable
.RDB_WR_A(RDB_WR_A), // 1-bit input: Port A read/write select
.RDB_WR_B(RDB_WR_B), // 1-bit input: Port B read/write select
.RST_A(RST_A), // 1-bit input: Port A asynchronous or synchronous reset for output
// registers

.RST_B(RST_B), // 1-bit input: Port B asynchronous or synchronous reset for output
// registers

.SLEEP(SLEEP) // 1-bit input: Dynamic power gating control
);
// End of URAM288_BASE_inst instantiation
    
```

## For More Information

- See the *UltraScale Architecture Memory Resources User Guide* ([UG573](#)).
- See the [UltraScale Programmable Devices Documentation](#).

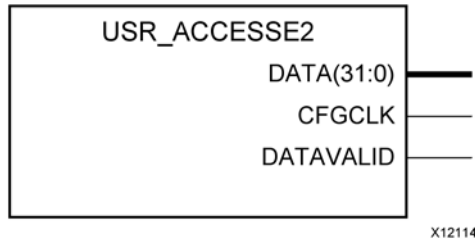
## USR\_ACESSE2

Primitive: Configuration Data Access

PRIMITIVE\_GROUP: CONFIGURATION

PRIMITIVE\_SUBGROUP: USR\_ACCESS

Families: UltraScale, UltraScale+



### Introduction

The USR\_ACESSE2 design element enables access to the 32-bit AXSS register within the configuration logic. This enables device logic to access static data that can be set from the bitstream. The primitive and functionality for the Kintex UltraScale and Virtex UltraScale FPGAs are identical to that for the 7 series. The USR\_ACESSE2 register AXSS can be used to provide a single 32-bit constant value to the device logic. The register contents can be defined during bitstream generation, avoiding the need to re-compile the design as would be required if distributed RAM was used to hold the constant. A constant can be used to track the version of the design, or any other information you require.

### Port Descriptions

Port	Direction	Width	Function
CFGCLK	Output	1	Configuration Clock
DATA<31:0>	Output	32	Configuration Data reflecting the contents of the AXSS register
DATAVALID	Output	1	Active High Data Valid

### Design Entry Method

Instantiation	Recommended
Inference	No
IP and IP Integrator Catalog	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_ACESSE2: Configuration Data Access
--           UltraScale
-- Xilinx HDL Libraries Guide, version 2016.1

USR_ACESSE2_inst : USR_ACESSE2
port map (
  CFGCLK => CFGCLK,      -- 1-bit output: Configuration Clock
  DATA => DATA,        -- 32-bit output: Configuration Data reflecting the contents of the AXSS register
  DATAVALID => DATAVALID -- 1-bit output: Active High Data Valid
);

-- End of USR_ACESSE2_inst instantiation
  
```

## Verilog Instantiation Template

```

// USR_ACESSE2: Configuration Data Access
//           UltraScale
// Xilinx HDL Libraries Guide, version 2016.1

USR_ACESSE2 USR_ACESSE2_inst (
  .CFGCLK(CFGCLK),      // 1-bit output: Configuration Clock
  .DATA(DATA),         // 32-bit output: Configuration Data reflecting the contents of the AXSS register
  .DATAVALID(DATAVALID) // 1-bit output: Active High Data Valid
);

// End of USR_ACESSE2_inst instantiation
  
```

## For More Information

- See the *UltraScale Architecture Configuration User Guide* ([UG570](#)).
- See the [UltraScale Programmable Devices Documentation](#).

# Additional Resources and Legal Notices

---

## Xilinx Resources

For support resources such as Answers, Documentation, Downloads, and Forums, see [Xilinx Support](#).

---

## Solution Centers

See the [Xilinx Solution Centers](#) for support on devices, software tools, and intellectual property at all stages of the design cycle. Topics include design assistance, advisories, and troubleshooting tips.

---

## Please Read: Important Legal Notices

The information disclosed to you hereunder (the "Materials") is provided solely for the selection and use of Xilinx products. To the maximum extent permitted by applicable law: (1) Materials are made available "AS IS" and with all faults, Xilinx hereby **DISCLAIMS ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE**; and (2) Xilinx shall not be liable (whether in contract or tort, including negligence, or under any other theory of liability) for any loss or damage of any kind or nature related to, arising under, or in connection with, the Materials (including your use of the Materials), including for any direct, indirect, special, incidental, or consequential loss or damage (including loss of data, profits, goodwill, or any type of loss or damage suffered as a result of any action brought by a third party) even if such damage or loss was reasonably foreseeable or Xilinx had been advised of the possibility of the same. Xilinx assumes no obligation to correct any errors contained in the Materials or to notify you of updates to the Materials or to product specifications. You may not reproduce, modify, distribute, or publicly display the Materials without prior written consent. Certain products are subject to the terms and conditions of Xilinx's limited warranty, please refer to Xilinx's Terms of Sale which can be viewed at <http://www.xilinx.com/legal.htm#tos>; IP cores may be subject to warranty and support terms contained in a license issued to you by Xilinx. Xilinx products are not designed or intended to be fail-safe or for use in any application requiring fail-safe performance; you assume sole risk and liability for use of Xilinx products in such critical applications, please refer to Xilinx's Terms of Sale which can be viewed at <http://www.xilinx.com/legal.htm#tos>.

© Copyright 2012–2016 Xilinx, Inc. Xilinx, the Xilinx logo, Artix, ISE, Kintex, Spartan, Virtex, Vivado, Zynq, and other designated brands included herein are trademarks of Xilinx in the United States and other countries. All other trademarks are the property of their respective owners.