Versal ACAP Memory Resources

Architecture Manual

AM007 (v1.0) July 16, 2020

XILINX®

Revision History

The following table shows the revision history for this document.

Section	Revision Summary
07/16/2020 Version 1.0	
Initial release.	N/A

Table of Contents

Chapter 1

Overview

Introduction to Versal ACAP

Versal™ adaptive compute acceleration platforms (ACAPs) combine Scalar Engines, Adaptable Engines, and Intelligent Engines with leading-edge memory and interfacing technologies to deliver powerful heterogeneous acceleration for any application. Most importantly, Versal ACAP hardware and software are targeted for programming and optimization by data scientists and software and hardware developers. Versal ACAPs are enabled by a host of tools, software, libraries, IP, middleware, and frameworks to enable all industry-standard design flows.

Built on the TSMC 7 nm FinFET process technology, the Versal portfolio is the first platform to combine software programmability and domain-specific hardware acceleration with the adaptability necessary to meet today's rapid pace of innovation. The portfolio includes six series of devices uniquely architected to deliver scalability and AI inference capabilities for a host of applications across different markets—from cloud—to networking—to wireless communications—to edge computing and endpoints.

The Versal architecture combines different engine types with a wealth of connectivity and communication capability and a network on chip (NoC) to enable seamless memory-mapped access to the full height and width of the device. Intelligent Engines are SIMD VLIW AI Engines for adaptive inference and advanced signal processing compute, and DSP Engines for fixed point, floating point, and complex MAC operations. Adaptable Engines are a combination of programmable logic blocks and memory, architected for high-compute density. Scalar Engines, including Arm® Cortex™-A72 and Cortex-R5F processors, allow for intensive compute tasks.

The Versal AI Core series delivers breakthrough AI inference acceleration with AI Engines that deliver over 100x greater compute performance than current server-class of CPUs. This series is designed for a breadth of applications, including cloud for dynamic workloads and network for massive bandwidth, all while delivering advanced safety and security features. AI and data scientists, as well as software and hardware developers, can all take advantage of the high-compute density to accelerate the performance of any application.

The Versal Prime series is the foundation and the mid-range of the Versal platform, serving the broadest range of uses across multiple markets. These applications include 100G to 200G networking equipment, network and storage acceleration in the Data Center, communications test equipment, broadcast, and aerospace & defense. The series integrates mainstream 58G transceivers and optimized I/O and DDR connectivity, achieving low-latency acceleration and performance across diverse workloads.

The Versal Premium series provides breakthrough heterogeneous integration, very high-performance compute, connectivity, and security in an adaptable platform with a minimized power and area footprint. The series is designed to exceed the demands of high-bandwidth, compute-intensive applications in wired communications, data center, test & measurement, and other applications. Versal Premium series ACAPs include 112G PAM4 transceivers and integrated blocks for 600G Ethernet, 600G Interlaken, PCI Express® Gen5, and high-speed cryptography.

The Versal architecture documentation suite is available at: https://www.xilinx.com/versal.

Navigating Content by Design Process

Xilinx® documentation is organized around a set of standard design processes to help you find relevant content for your current development task. This document covers the following design processes:

- **System and Solution Planning:** Identifying the components, performance, I/O, and data transfer requirements at a system level. Includes application mapping for the solution to PS, PL, and AI Engine. Topics in this document that apply to this design process include:
  - Block RAM Introduction
  - Synchronous Dual-Port and Single-Port RAMs
  - Additional Block RAM Features
  - UltraRAM Introduction
  - UltraRAM Key Features
  - UltraRAM Cascade
  - Block RAM and UltraRAM Differences
  - Additional Memory Resources
- **Hardware, IP, and Platform Development:** Creating the PL IP blocks for the hardware platform, creating PL kernels, subsystem functional simulation, and evaluating the Vivado® timing, resource use, and power closure. Also involves developing the hardware platform for system integration. Topics in this document that apply to this design process include:
  - Design Entry Methods
  - Differences from Previous Generations
  - Block RAM Library Primitives

- Additional RAMB18E5 and RAMB36E5 Primitive Design Considerations
- UltraRAM Primitives
- Additional URAM288E5 Primitive Design Considerations
- UltraRAM Timing Diagrams

Block RAM

Block RAM Introduction

In addition to distributed RAM and high-speed SelectIO™ memory interfaces, Versal devices feature a large number of 36 Kb block RAMs. Each 36 Kb block RAM contains two independently controlled 18 Kb RAMs. Block RAMs are placed in columns within the clock regions (CRs) and across the device. The block RAM blocks are cascadable to enable a deeper memory implementation, have a sleep mode for power savings, and have selectable write mode operations. For more information about clock regions, see *Versal ACAP Clocking Resources Architecture Manual* (AM003).

Block RAM Summary

The block RAM in Versal devices stores up to 36 Kbits of data and can be configured as either two independent 18 Kb RAMs, or one 36 Kb RAM. Each block RAM has two write and two read ports. A 36 Kb block RAM can be configured with independent port widths for each of those ports as 4K x 9, 2K x 18, or 1K x 36 (when used as true dual-port (TDP) memory). If only one write and one read port are used, a 36 Kb block RAM can additionally be configured with a port width of 512 x 72 bits (when used as simple dual-port (SDP) memory). An 18 Kb block RAM can be configured with independent port widths for each of those ports as 2K x 9 or 1K x 18 (when used as TDP memory). If only one write and one read port are used, an 18 Kb block RAM can additionally be configured with a port width of 512 x 36 bits (when used as SDP memory).

Identical to the UltraScale series FPGA block RAMs, write and read are synchronous operations. The two ports are symmetrical and totally independent, sharing only the stored data. Each port can be configured in one of the available widths, independent of the other port. In addition, the read port width can be different from the write port width for each port. The memory content can be initialized or cleared by the configuration bitstream. During a write operation, the memory can be set to have the data output remain unchanged, reflect the new data being written or the previous data now being overwritten.

The block RAM features include:

- Per-block memory storage capability where each block RAM can store up to 36 Kbits of data.

- Support of two independent 18 Kb blocks, or a single 36 Kb block RAM.
- Each 36 Kb block RAM can be used with a single read and write port (SDP), doubling data width of the block RAM to 72 bits. The 18 Kb block RAM can also be used with a single read and write port, doubling data width to 36 bits.
- When used as RAMB36 SDP memory, one port width is fixed (i.e., 512 x 64 or 512 x 72). The other port width can then be 4K x 9 through 512 x 72. When used as RAMB18 SDP memory, one port width is fixed (i.e., 512 x 36). The other port width can then be 2K x 9 through 512 x 36.
- The data outputs of the lower to upper adjacent block RAMs can be cascaded to build large block RAM blocks. Optional pipeline registers are available to support maximum performance.
- One 64-bit error correction coding (ECC) block is provided per 36 Kb block RAM. Independent encode/decode functionality is available. ECC mode has the capability of injecting errors.
- Synchronous or asynchronous set/resets of the outputs to an initial value is available for both the latch and register modes of the block RAM output.
- Separate synchronous or asynchronous set/reset pins independently control the set/reset of the optional output registers and output latch stages in the block RAM.
- 18, 36, or 72-bit wide block RAM ports can have an individual write enable per byte. There is also an additional byte write enable for the parity bits. This feature is popular for interfacing to a microprocessor.
- All inputs are registered with the port clock and have a setup-to-clock timing specification.
- All outputs have a read function or a read-during-write function, depending on the state of the write enable (WE) pin. The outputs are available after the clock-to-out timing interval. The read-during-write outputs have one of three operating modes: WRITE_FIRST, READ_FIRST, and NO_CHANGE.
- A write operation requires one clock edge.
- A read operation requires one clock edge.
- All output ports are latched or registered (optional). The state of the output port does not change until the port executes another read or write operation. The default block RAM output is register mode.

UltraRAM

UltraRAM Introduction

UltraRAM is a single-clocked, two port, synchronous memory available in Versal devices. Because UltraRAM is compatible with the columnar architecture, multiple UltraRAMs can be instantiated and directly cascaded in an UltraRAM column for the entire height of the device. A column in a single clock region contains 24 UltraRAM blocks. Devices with UltraRAM include multiple UltraRAM columns distributed in the device. Most of the devices in the Versal architecture include UltraRAM blocks. For the available quantity of UltraRAM in specific devices, see *Versal Architecture and Product Data Sheet: Overview* (DS950).

UltraRAM Summary

UltraRAM blocks are 288 Kb, single-clock, synchronous memory blocks arranged in one or more columns in the device. There are 24 UltraRAM blocks per clock region per column. Multiple UltraRAM blocks can be cascaded together within a column using dedicated cascade routing, and the only limit is the height of the device or a single super logic region (SLR) in a stacked silicon interconnect (SSI) device. In addition, multiple columns can be cascaded together using a small quantity of logic resources. There is no timing penalty with cascading UltraRAM blocks if they are appropriately pipelined.

UltraRAM is a flexible, high-density memory building block. Each UltraRAM block can store up to 288K bits of data and is configured as a 4K x 72 memory block. UltraRAM has eight times the capacity of a block RAM. Similar to the block RAM, there are multiple UltraRAM columns distributed on the device. UltraRAM has two ports, both of which address all 4K x 72 bits. Each port can independently perform either one read or one write operation per clock cycle per port. However, internally the SRAM array uses single port memory cells. Dual port operation is achieved by executing port A operation followed by port B operation in a single cycle. Therefore, both ports share a single clock input. Each port can only execute either a write or read operation in one cycle. When executing a write operation, the read outputs are unchanged and hold the previous value.

The 288 Kb blocks can be cascaded to facilitate deeper memory implementations. Most of the routing related to cascading is contained inside the UltraRAM columns. Therefore, very little or no general interconnect is required and timing penalties are not incurred due to routing if the UltraRAM blocks are appropriately pipelined.

UltraRAM contains up to four pipeline stages for each of the two port interfaces. In a standalone, non-cascaded mode, the UltraRAM can be configured for one to four clock cycles latency, though typically, only one to three cycles of latency are required, depending on the target frequency. Cascade mode latency is a function of the size of the UltraRAM chain, frequency target, and other constraints. Similarly, clock-to-out performance depends on the selected output registers. Use the Vivado® tools to determine the performance and clock-to-out timing for specific design implementations.

Additional Memory Resources

In addition to block RAMs and UltraRAMs, the Versal ACAP includes small 64-bit RAMs distributed throughout the programmable logic, specialized RAMs in the processing system, and integrated memory controllers for access to external DDR memories. Details of these additional memory resources are available in the following documents:

- See the *Versal ACAP Configurable Logic Block Architecture Manual* (AM005) for small (64-bit) distributed RAMs in the programmable logic.
- See the *Versal ACAP Technical Reference Manual* (AM011) for processing system on-chip memory (OCM), tightly coupled memories (TCM), accelerator RAM (XRAM), and controller interfaces to external non-volatile memories. The OCM and XRAM can be accessed from the programmable logic through AXI interfaces.
- See the *Versal ACAP Programmable Network on Chip and Integrated Memory Controller LogiCORE IP Product Guide* (PG313) for the integrated memory controller interface to external DDR memories.

See the *Versal Architecture and Product Data Sheet: Overview* (DS950) for availability and quantities of block RAM, UltraRAM, and additional memory resources in each Versal ACAP.

Differences from Previous Generations

Block RAM

- Support for bit widths x1, x2, x4 has been removed.
- The default port width has changed to 0 for RAMB36 and RAMB18.
- Hard FIFO support is removed. Fabric logic can be used to implement FIFO functions.
- Address enable/compare feature is removed.
- Systolic cascade support has been removed.

- SRVAL/INIT attributes are merged into a single attribute.
- Async reset on the output registers is supported.
- Byte write mode parity interleaved and parity independent (x72).
- The ADDREN pin has been removed.
- The CASDIMUX pins have been removed.
- The ECCPARITY[7:0] pins have been removed.
- RDADDRECC[8:0] has been removed.
- Support for an asynchronous reset to logic 0 for the output register has been added:
    - A new attribute RST_MODE has been added to support this feature.
    - Additional pins ARST_A and ARST_B have been added.
- An extra bit WEBWE[8] was added to the WEBWE bus to support a separate parity byte write enable in x72 mode:
    - A new attribute BWE_MODE_B supports this feature.

UltraRAM Block

- Asymmetric ports widths are supported by hardware:
    - New aspect ratios have been added in hardware in addition to 4K x 72 (with ECC): 8K x 36, 16K x 18, and 32K x 9.
- Memory content can be initialized to user-defined values.
- Data cascade ordering has been added. There are two cascade order attributes to allow for separate control of control signals and data.

Design Entry Methods

Memories and FIFOs in the programmable logic can be constructed with block RAMs within the Vivado IP integrator block design flow, block RAMs can be inferred during high-level synthesis or synthesis of VHDL or Verilog code, or block RAMs can be explicitly instantiated and initialized in VHDL or Verilog code.

Single- or dual-port memories can be constructed with UltraRAMs that are instantiated and initialized to user-defined values in VHDL or Verilog code. UltraRAMs can also be inferred during synthesis from VHDL or Verilog code. See the *Vivado Design Suite User Guide: Synthesis* (UG901) for RAM HDL coding techniques.

Chapter 2

Block RAM Resources

Synchronous Dual-Port and Single-Port RAMs

Data Flow

The true dual-port 36 Kb block RAM dual-port memories consist of a 36 Kb storage area and two completely independent access ports, A and B. Similarly, each 18 Kb block RAM dual-port memory consists of an 18 Kb storage area and two completely independent access ports, A and B. The structure is fully symmetrical, and both ports are interchangeable. The following figure illustrates the true dual-port data flow of a RAMB36. The following table lists the port functions and descriptions.

Data can be written to either or both ports and can be read from either or both ports. Each write operation is synchronous, and each port has its own address, data in, data out, clock, clock enable, and write enable. The read and write operations are synchronous and require a clock edge.

There is no dedicated monitor to arbitrate the effect of identical addresses on both ports.

IMPORTANT! *The two clocks must be timed appropriately. Conflicting simultaneous writes to the same location never cause any physical damage but can result in data uncertainty.*

Note: The Vivado® tools automatically determine if a block RAM is used in SDP or TDP mode.

Figure 1: **RAMB36 Usage in a True Dual-Port Data Flow**

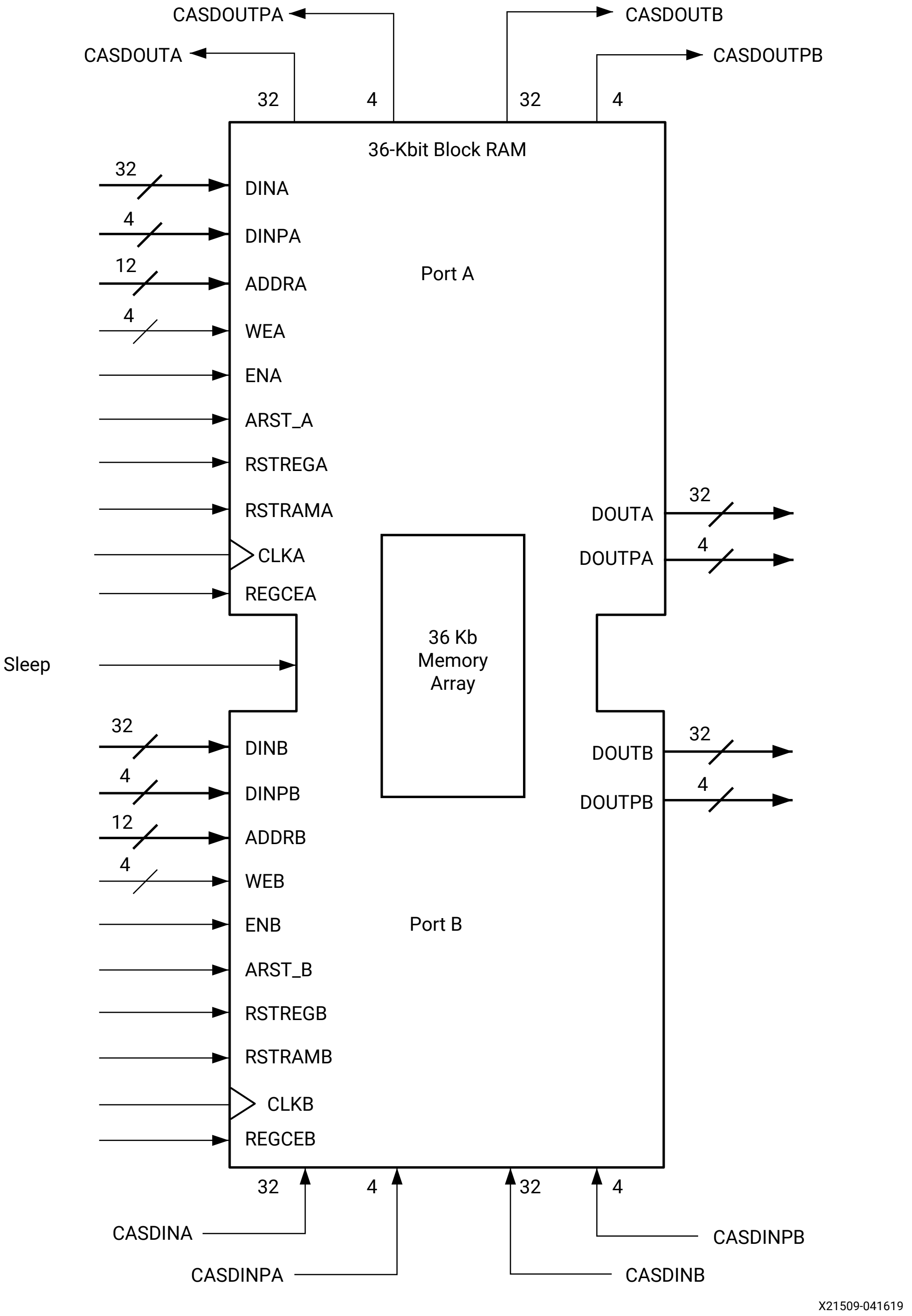


Table 1: **True Dual-Port Functions and Descriptions**

Port Function	Description
ARST_[A|B]	Asynchronous reset that resets the output register for port A and B to all zeros.
DIN[A|B]	Data input bus.
DINP[A|B][1]	Data input parity bus. Can be used for additional data inputs.
ADDR[A|B]	Address bus.
WE[A|B]	Byte-wide write enable.
EN[A|B]	When inactive, no data is written to the block RAM and the output bus remains in its previous state.
RSTREG[A|B]	Synchronous set/reset of the output registers (DO_REG = 1). The RSTREG_PRIORITY attribute determines the priority over REGCE.
RSTRAM[A|B]	Synchronous set/reset of the output data latches.
CLK[A|B]	Clock input.
DOUT[A|B]	Data output bus.
DOUTP[A|B][1]	Data output parity bus. Can be used for additional data outputs.
REGCE[A|B]	Output register clock enable.
CASDIN[A|B]	Cascade data input bus.
CASDINP[A|B]	Cascade parity input bus.
CASDOUT[A|B]	Cascade data output bus.
CASDOUTP[A|B]	Cascade parity output bus.
SLEEP	Dynamic shutdown power saving. If SLEEP is active, the block is in power saving mode.

Notes:

1. Data-In Buses – DINADIN, DINPADINP, DINBDIN, and DINPBDINP has more information on data parity pins.
2. Block RAM primitive port names can be different from the port function names.
3. For a more complete cascade data flow and port descriptions, see Cascadable Block RAM and Block RAM Library Primitives.

Read Operation

In latch mode, the read operation uses one clock edge. The read address is registered on the read port, and the stored data is loaded into the output latches after the RAM access time. When using the output register, the read operation takes one extra latency cycle.

Write Operation

A write operation is a single clock-edge operation. The write address is registered on the write port, and the data input is stored in memory.

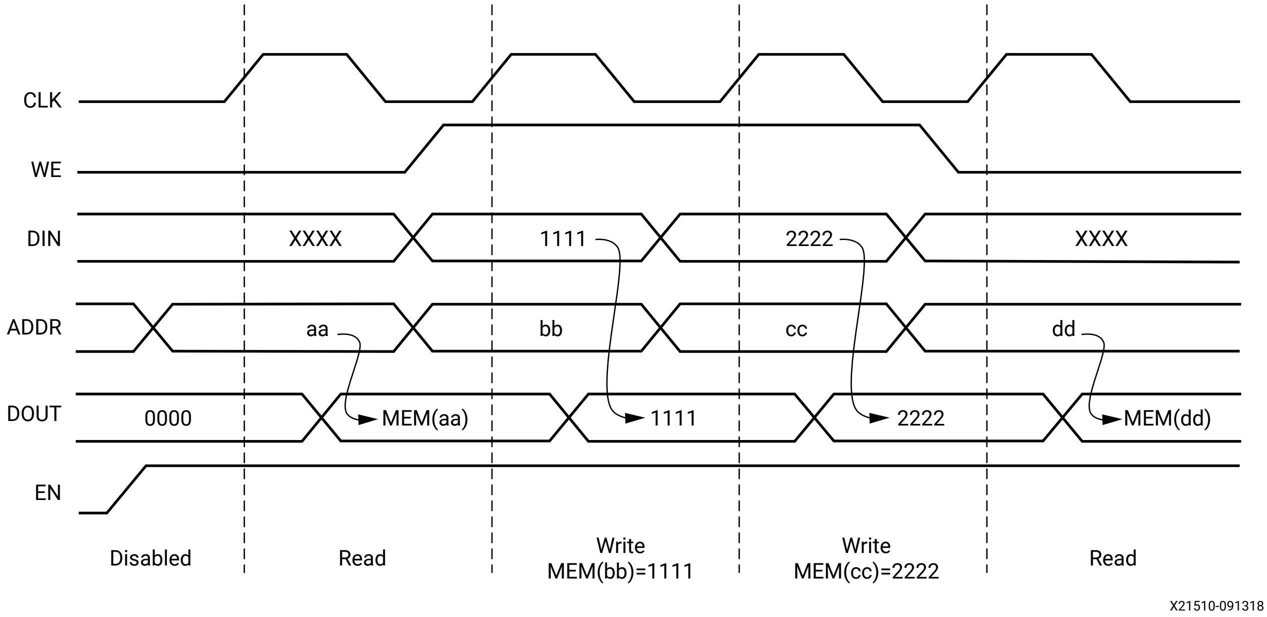
Write Modes

Three settings of the write mode determine the behavior of the data available on the output latches after a write clock edge: WRITE_FIRST, READ_FIRST, and NO_CHANGE. Write mode selection is set by configuration. The write mode attribute can be individually selected for each port. The default mode is NO_CHANGE. WRITE_FIRST outputs the newly written data onto the output bus. READ_FIRST outputs the previously stored data while new data is being written. NO_CHANGE maintains the output previously generated by a read operation.

WRITE_FIRST or Transparent Mode

In WRITE_FIRST mode, the input data is simultaneously written into memory and stored in the data output (transparent write), as shown in the following figure. These waveforms correspond to latch mode when the optional output pipeline register is not used.

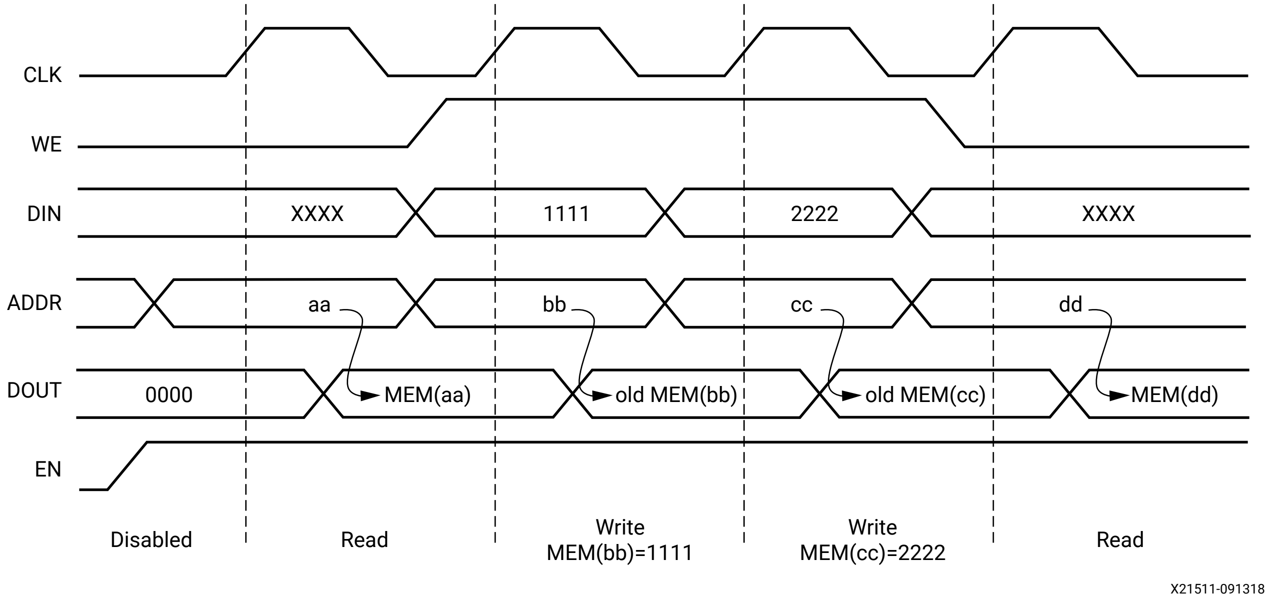
Figure 2: **WRITE_FIRST Mode Waveforms**



READ_FIRST or Read-Before-Write Mode

In READ_FIRST mode, data previously stored at the write address appears on the output latches while the input data is being stored in memory (read before write). The waveforms in the following figure correspond to latch mode when the optional output pipeline register is not used.

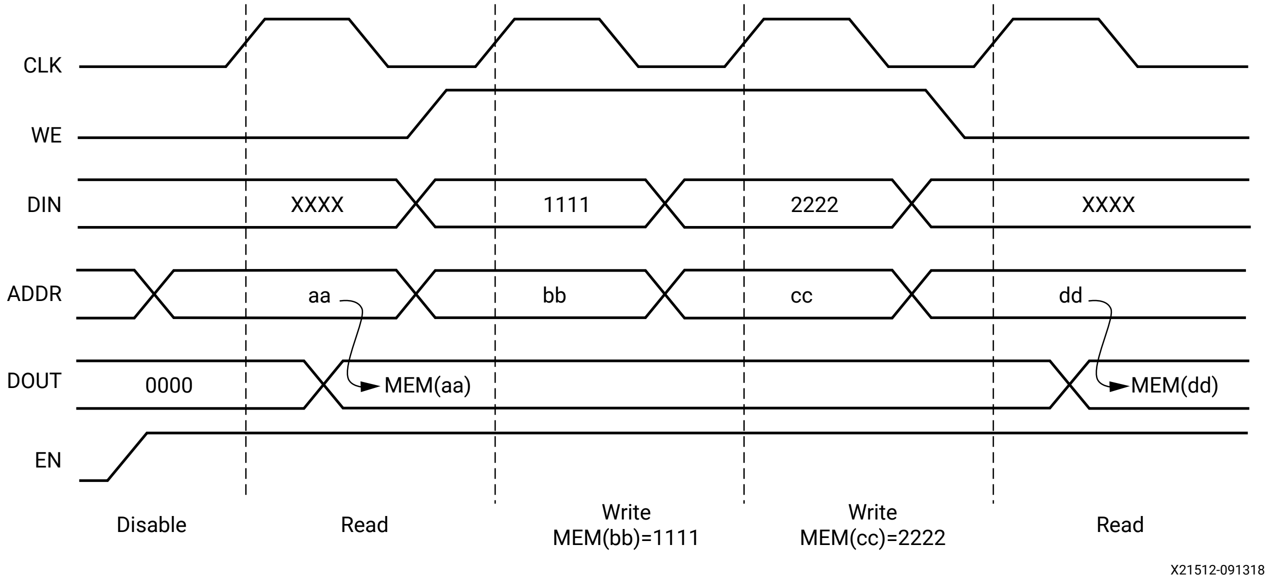
Figure 3: **READ_FIRST Mode Waveforms**



NO_CHANGE Mode (DEFAULT)

In NO_CHANGE mode, the output latches remain unchanged during a write operation. As shown in the following figure, data output remains the last read data and is unaffected by a write operation on the same port. These waveforms correspond to latch mode when the optional output pipeline register is not used. NO_CHANGE mode is the most power efficient.

Figure 4: **NO_CHANGE Mode Waveforms**



Address Collision

An address collision is when both block RAM ports access the same address location in the same clock cycle. There are two fundamental clock type setups: common clock and independent clock. Common (synchronous) clocks are driven by a common clock buffer driver. All other CLKA and CLKB connections are considered independent (asynchronous) clocks. The CLOCK_DOMAINS attribute must also be set appropriately. See Table 16 for legal and default values. If no address collisions are expected or possible (SDP configurations) to save power, the recommended write mode is NO_CHANGE. Using READ_FIRST mode has a power penalty over NO_CHANGE and should only be used when necessary for functionality or to address collision mitigation.

- When both ports are reading, the operations complete successfully.
- When both ports are writing different data, the memory location is written with non-deterministic data.
- When one port is writing and the other port is reading, the write is always successful but the resulting read memory value can vary. See the following tables.

Table 2: **Common Clock**

Clock Type	Write Mode Port A	Write Mode Port B	Write Enable Port A (Data)	Write Enable Port B (Data)	Resulting Data Out Port A	Resulting Data Out Port B	Resulting Memory Value
Common	RF/WF/NC	RF/WF/NC	0	0	Old memory data	Old memory data	No change
Common	RF	RF/WF/NC	1 (DIA)	0	Old memory data	Old memory data	DIA
Common	WF	RF/WF/NC	1 (DIA)	0	DIA	X	DIA
Common	NC	RF/WF/NC	1 (DIA)	0	No change	X	DIA
Common	RF/WF/NC	RF	0	1 (DIB)	Old memory data	Old memory data	DIB
Common	RF/WF/NC	WF	0	1 (DIB)	X	DIB	DIB
Common	RF/WF/NC	NC	0	1 (DIB)	X	No change	DIB
Common	RF/WF/NC	RF/WF/NC	1	1	X	X	X

Notes:

1. Common clocked access collision is when the port addresses are the same for the same clock cycle.

Table 3: **Independent Clock**

Clock Type	Write Mode Port A	Write Mode Port B	Write Enable Port A (Data)	Write Enable Port B (Data)	Resulting Data Out Port A	Resulting Data Out Port B	Resulting Memory Value
Independent	RF/WF/NC	RF/WF/NC	0	0	Old memory data	Old memory data	No change
Independent	RF	RF/WF/NC	1 (DIA)	0	Old memory data	X	DIA

Table 3: **Independent Clock** *(cont'd)*

Clock Type	Write Mode Port A	Write Mode Port B	Write Enable Port A (Data)	Write Enable Port B (Data)	Resulting Data Out Port A	Resulting Data Out Port B	Resulting Memory Value
Independent	WF	RF/WF/NC	1 (DIA)	0	DIA	X	DIA
Independent	NC	RF/WF/NC	1 (DIA)	0	No change	X	DIA
Independent	RF/WF/NC	RF	0	1 (DIB)	X	Old memory data	DIB
Independent	RF/WF/NC	WF	0	1 (DIB)	X	DIB	DIB
Independent	RF/WF/NC	NC	0	1 (DIB)	X	No change	DIB
Independent	RF/WF/NC	RF/WF/NC	1	1	X	X	X

Notes:

1. An independently clocked access collision might occur when the port addresses are the same and when the clock edges of the two ports are within the same clock cycle. The UNISIM can report an error during simulation for collisions when the SIM_COLLISION_CHECK attribute is set to ALL (default).

In Table 2 and Table 3:

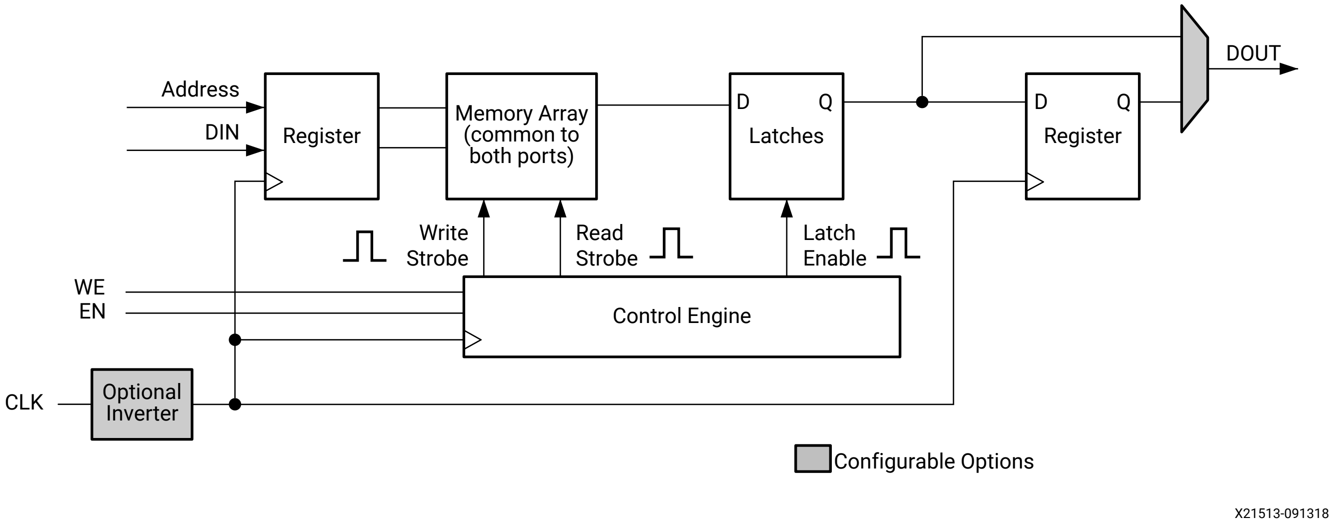
- Write enable is active-High, 1 = Write, 0 = Read
- RF = READ_FIRST, WF = WRITE_FIRST, NC = NO_CHANGE
- X = Undeterministic value
- DIA = Port A data input, DIB = Port B data input

Additional Block RAM Features

Optional Output Registers

The optional output registers improve design performance by eliminating routing delay to the configurable logic block (CLB) flip-flops for pipelined operation. An independent clock and clock enable input is provided for these output registers. As a result, the output data registers hold the value independent of the input register operation. The following figure shows the optional output register.

Figure 5: **Block RAM Logic Diagram (One Port Shown)**



Independent Read and Write Port Width Selection

Each block RAM port has control over data width and address depth (aspect ratio). The true dual-port block RAM extends this flexibility to read and write where each individual port can be configured with different data bit widths. For example, port A can have a 36-bit read width and a 9-bit write width, and port B can have an 18-bit read width and a 36-bit write width.

If the read port width differs from the write port width and is configured in WRITE_FIRST mode, DOUT shows valid new data for all the enabled write bytes. The DOUT port outputs the original data stored in memory for all not-enabled bytes.

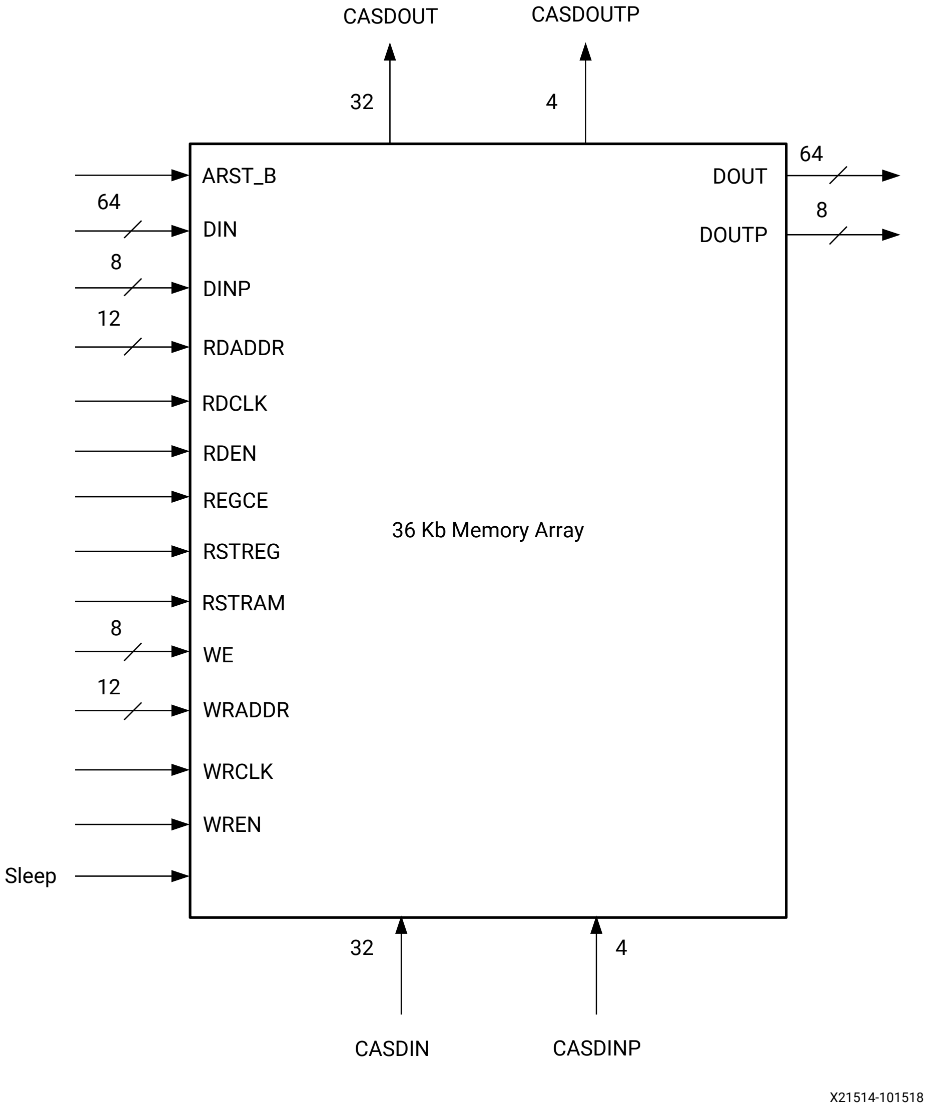
Independent read and write port width selection increases the efficiency of implementing a content addressable memory (CAM) in block RAM. This option is available for all Versal devices true dual-port RAM port sizes and modes.

Simple Dual-Port Block RAM

Each 18 Kb block and 36 Kb block can also be configured in a SDP RAM mode. In this mode, the block RAM port width doubles to 36 bits for the 18 Kb block RAM and 72 bits for the 36 Kb block RAM. When the block RAM is used as SDP memory, independent read and write operations can occur simultaneously, where port A is designated as the read port and port B as the write port. When the read and write port access the same data location at the same time, it is treated as a collision, identical to the port collision in true dual-port mode. Versal devices support these modes when the block RAM is used as SDP memory (READ_FIRST, WRITE_FIRST, NO_CHANGE).

The following figure shows the simple dual-port data flow for RAMB36 when the block RAM is used as SDP memory.

Figure 6: **RAMB36 Usage in a Simple Dual-Port Data Flow**



This table lists the simple dual-port functions and descriptions.

Table 4: **Simple Dual-Port Functions and Descriptions**

Port Function	Description
ARST_B	Asynchronous reset that resets the output register DOUT to all zeros.
DOUT	Data output bus.
DOUTP	Data output parity bus.
DIN	Data input bus.
DINP	Data input parity bus.
RDADDR	Read data address bus.
RDCLK	Read data clock.

Table 4: **Simple Dual-Port Functions and Descriptions** *(cont'd)*

Port Function	Description
RDEN	Read port enable.
REGCE	Output register clock enable.
RSTREG	Synchronous set/reset of the output registers.
RSTRAM	Synchronous set/reset of the output data latches.
WRADDR	Write data address bus.
WRCLK	Write data clock.
WREN	Write port enable.
SLEEP	Dynamic shutdown power saving. If Sleep is High, the block is in power-saving mode.
CASDIN[A|B]	Cascade data input bus.
CASDINP[A|B]	Cascade parity input bus.
CASDOUT[A|B]	Cascade data output bus.
CASDOUTP[A|B]	Cascade parity output bus.

Notes:

1. For a more complete cascade data flow and port descriptions, see Cascadable Block RAM and Block RAM Library Primitives.

Cascadable Block RAM

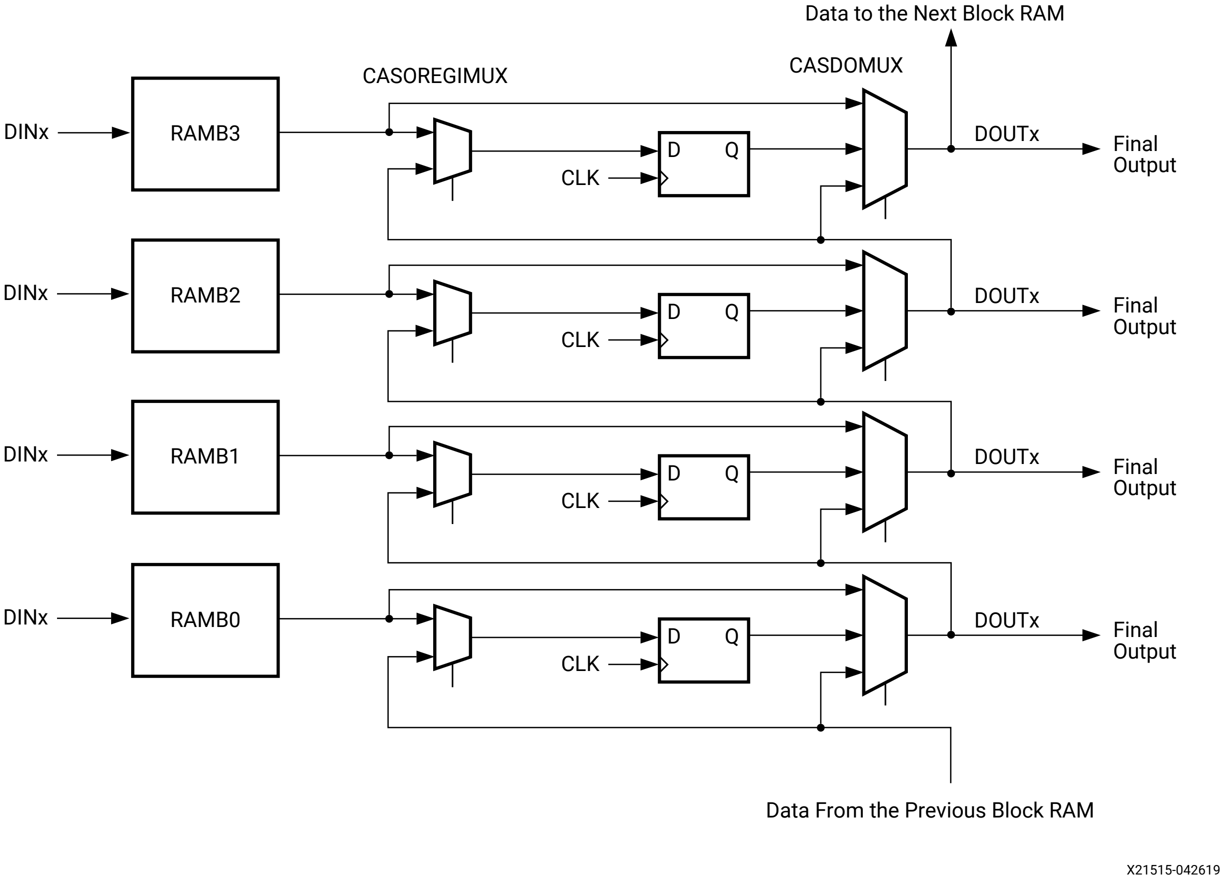
Versal devices provide the capability to cascade data out from one RAMB36 to the next RAMB36 serially to make a deeper block RAM in a bottom-up fashion. The data out cascading feature is supported for all RAMB36 port widths. The block RAM cascade supports all the features supported by the RAMB36E5 module.

The data flow is always from lower block RAM to upper block RAM. All of the signal routings and the control logic for the cascading feature are implemented in hardware. Multiple block RAMs can be cascaded, as required. In cascade mode, a single, common clock source must drive the same block RAM inputs (RDCLK or WRCLK). Furthermore, the data cascade capability allows that the lower RAMB18 of the lower RAMB36 can be independently cascaded to the lower RAMB18 of the upper RAMB36. Similarly, the upper RAMB18 of lower RAMB36 can be cascaded to the upper RAMB18 of the upper RAMB36 site.

***Note*:** All block RAMs in a cascade chain must have matching configurations for certain features (for example, common inputs such as the port width must be identical).

The following figure shows a high-level, conceptual view of four cascaded block RAMs.

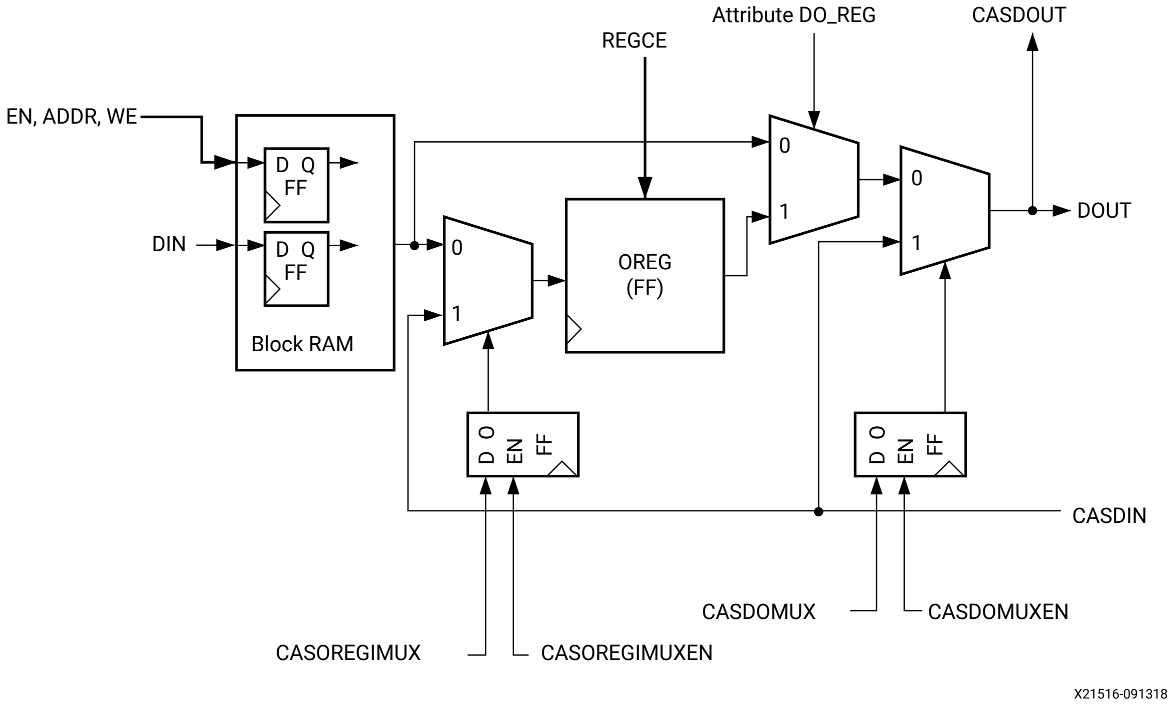
Figure 7: **High-Level View of the Block RAM Cascade Architecture**



The block RAM provides flexibility to support many different implementations of the cascade feature. The three multiplexers (see figure) that select datapaths and pipeline registers can be dynamically controlled with the input pins.

The following figure shows a more detailed diagram of the functional implementation in a single block RAM block. Two cascade multiplexer selection pins are available when the block RAM is in cascade mode. CASOREGIMUX selects the data output of the block RAM or the cascaded data input to the block RAM's optional output register. This control pin allows pipelined cascading for maximum performance. CASDOMUX selects the data output of the block RAM (with or without the optional register) or the cascaded data input. The latter two cascade multiplexer select pins are registered at the input and have an enable control pin. CASDOUT and CASDIN have dedicated interconnects within a block RAM column. Both the cascade connections and data connection to and from the block RAM are available at the same time.

Figure 8: **Cascade Functional Diagram**



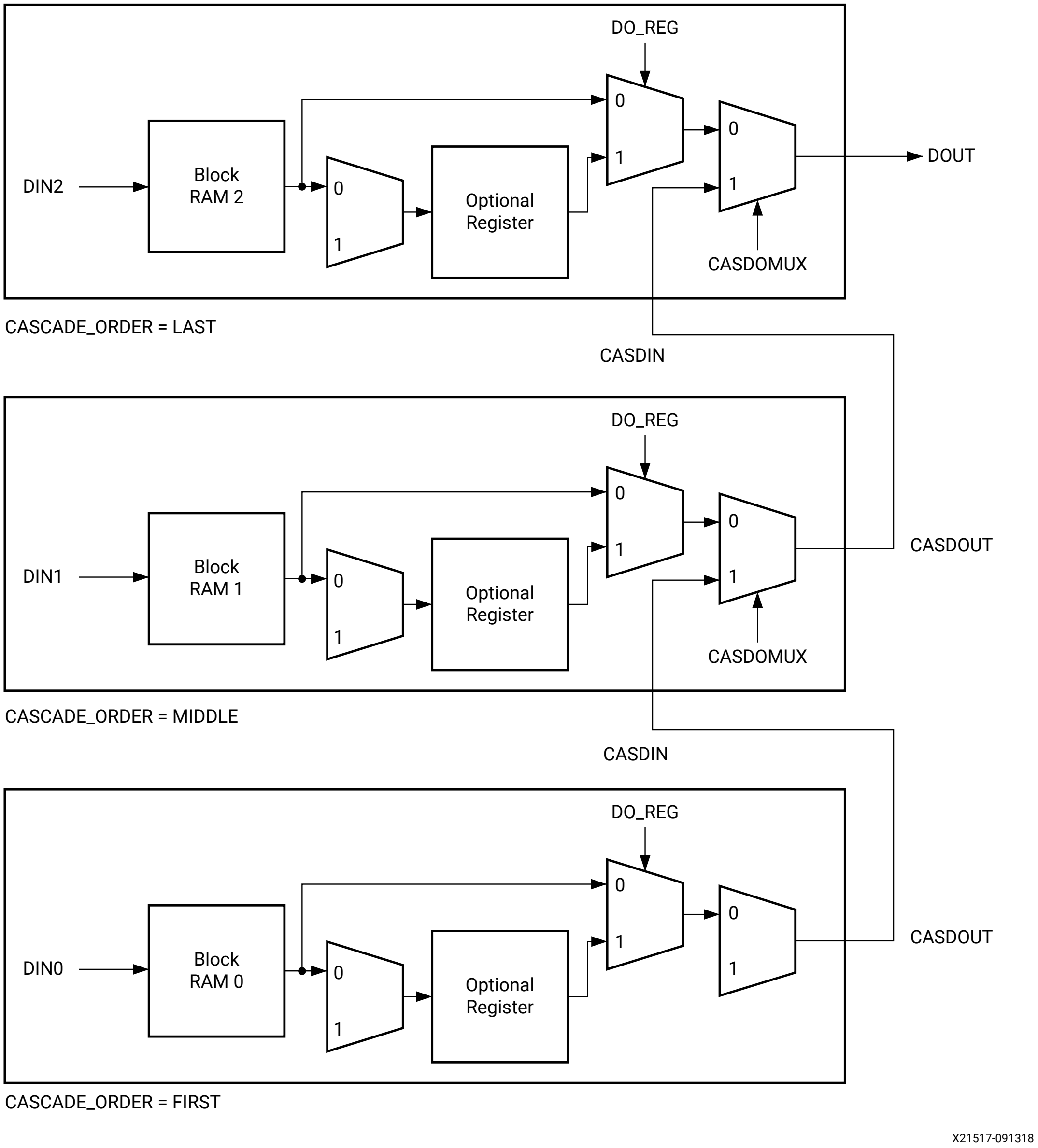
Although many different use cases can be implemented using the block RAM data cascade feature, this chapter describes two of the most common use cases. The examples shown are based on cascading three block RAM blocks, but more block RAM blocks can be cascaded with some limitations as required by the application in the same fashion.

Standard Data Output Cascade Mode

In this cascade use case, the data out of the lower block RAM is multiplexed to the final output multiplexer of the upper block RAM (see the following figure). The cascading can be applied to an entire block RAM column. This case yields a very deep RAM that can be implemented using only a few logic resources that might be required to drive the EN pins, drive the pins of the block RAM, determine the correct select value for the cascade muxes, and align the data if the DO_REG is used. The input multiplexer always selects DIN to write to the block RAM, the block RAM output multiplexer always selects the block RAM output data, and the last output multiplexer selects the current block RAM data (optionally registered) or the cascaded data from the block RAM below. The length of the block RAM chain impacts the final clock-to-out performance, which might slow down the performance depending on how many block RAMs are cascaded. All features of the block RAM are supported.

Note: The attribute CASCADE_ORDER defines the placement sequence within a block RAM column while the DO_REG attribute turns the optional block RAM register on or off.

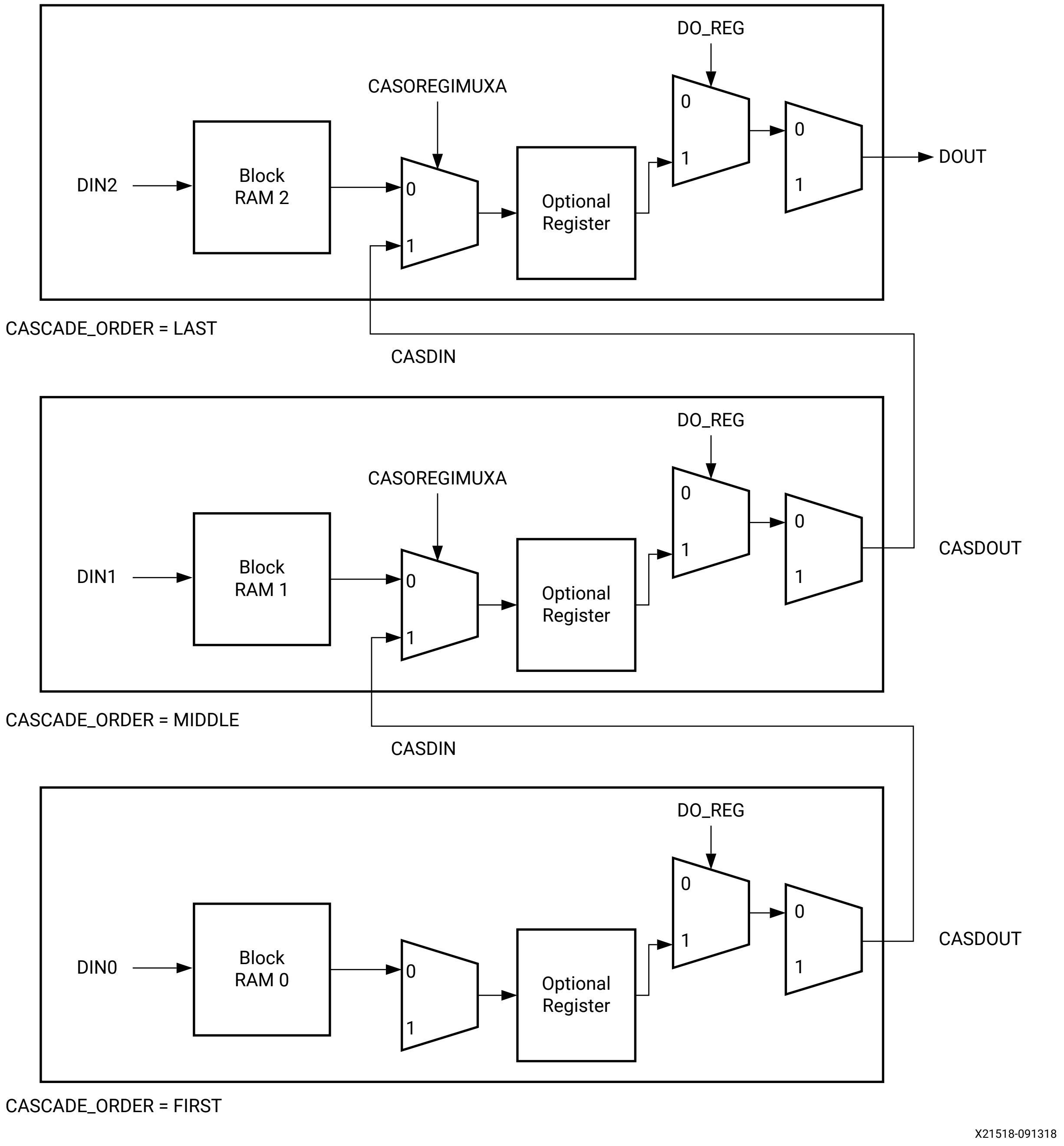
Figure 9: **Block RAM Cascade – Standard Data Out Cascade**



Data Out Cascade in Pipeline Mode

The block RAM pipeline cascade mode is similar to the standard data output cascade mode but allows the application to use the cascade mode at higher frequencies (see the following figure). The cascading data output propagates through the regular block RAM output registers because they are used as additional pipeline stages to achieve higher frequencies in this cascade mode. The external CASOREGIMUX pin controls the multiplexer that selects the input to the optional register. Thus, the data from the block RAM below or the current block RAM can be stored into the output register. The input multiplexer always selects DIN to write to the block RAM, the block RAM output multiplexer selects the block RAM output data, or the cascaded data from the block RAM below to write to the register. The final output multiplexer for each of the cascade stages always selects the data from the register for the final output data. All the DO_REG attributes have to be set to TRUE in this case. In this cascading mode, the length of the cascade chain is limited to within one clock region.

Figure 10: **Block RAM Cascade – Pipelined Data Out Cascade**



Byte-Wide Write Enable

The byte-wide write enable feature of the block RAM enables the writing of nine-bit (one byte) portions of incoming data. There are four independent byte-wide write enable inputs to the RAMB36E5 true dual-port RAM. In TDP mode for RAMB36E5, there are two ports, A and B, each of which have a 4-bit write enable bus (one bit corresponding to each data byte). In SDP mode for RAMB36E5, there is one write port, which has a 9-bit write enable bus (one bit corresponding to each data byte, one bit for parity). In RAMB36SDP mode, the user can choose between two modes for the attribute BWE_MODE_B. In PARITY_INTERLEAVED mode, each write enable bit enables eight DIN bits and one DINP. In the PARITY_INDEPENDENT mode, each write enable bit enables eight DIN bits, and the extra enable bit controls all eight DINP bits. By default, this attribute is set to PARITY_INTERLEAVED. Each byte-wide write enable is associated with one byte of input data and one parity bit or eight DIN bytes and an independent bit for parity depending on whether the BWE_MODE_B is set to PARITY_INTERLEAVED or PARITY_INDEPENDENT. The PARITY_INDEPENDENT case only applies to the SDP mode for a width of 72. The byte-wide write enable inputs must be driven in accordance with the data width configurations and parity bit usage. This feature is useful when using block RAM to interface with a microprocessor. Byte-wide write enable is not available in the ECC mode. Byte-wide write enable is further described in the Additional RAMB18E5 and RAMB36E5 Primitive Design Considerations. The following figure shows the byte-wide write enable timing diagram for the RAMB36E5.

Figure 11: **Byte-wide Write Operation Waveforms (x36 WRITE_FIRST)**

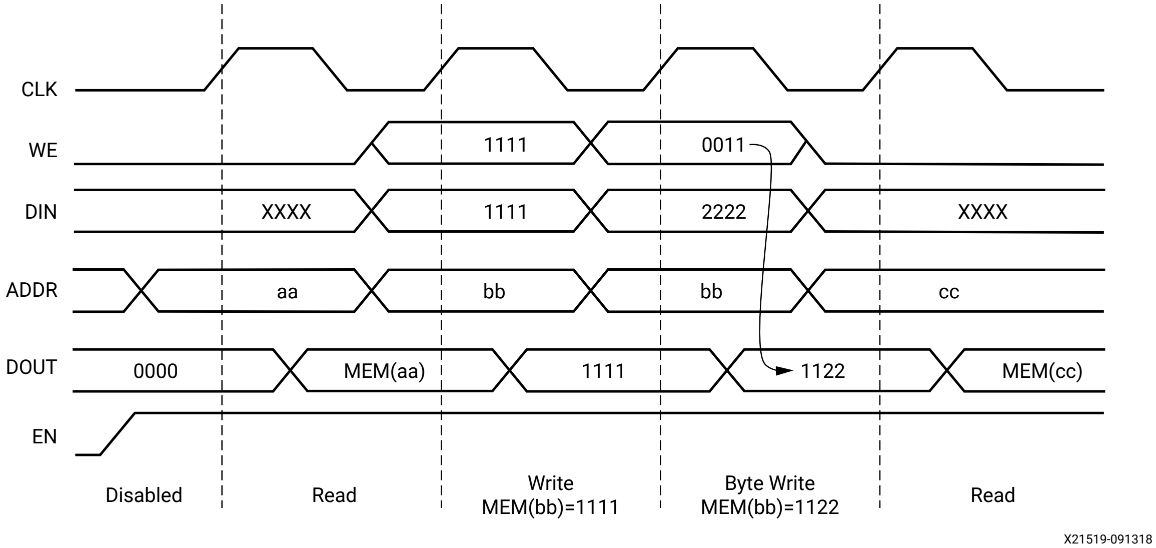


Table 5: **Available Byte-wide Write Enables**

Primitive	BWE_MODE_B	Maximum Bit Width	Bits of WE	Comments
RAMB36E5 TDP mode	N/A	36	WEA/B<3:0>	WEA/B<3:0> each controls 8 DIN+1 DIP

Table 5: **Available Byte-wide Write Enables** *(cont'd)*

Primitive	BWE_MODE_B	Maximum Bit Width	Bits of WE	Comments
RAMB36E5 SDP mode	PARITY_INTERLEAVED (default)	72	WEB<7:0>	WEB<7:0> each controls 8 DIN + 1 DIP
	PARITY_INDEPENDENT	72	WEB<8:0>	WEB<7:0> controls 8DIN WEB<8> controls 8 DIP
RAMB18E5 TDP mode	N/A	18	WEA/B<1:0>	WEA/B<1:0> each controls 8 DIN + 1 DIP
RAMB18E5 SDP mode	N/A	36	WEB<3:0>	WEB<3:0> each controls 8 DIN + 1 DIP

When the RAMB36E5 is configured for a 36-bit or 18-bit wide datapath, any port can restrict writing to specified byte locations within the data word. If configured in READ_FIRST mode, the DOUT bus shows the previous content of the whole addressed word. In WRITE_FIRST mode, DOUT shows a combination of the newly written enabled byte(s), and the initial memory contents of the unwritten bytes.

Block RAM Error Correction Code

The block RAM implementation of the 36 Kb block RAM supports a 64-bit ECC implementation. The code is used to detect single- and double-bit errors in block RAM data read out. Single-bit errors are then corrected in the output data.

Power Gating of Unused Block RAMs

Versal devices power down unused/uninstantiated block RAM blocks at an 18 Kb granularity. Power gating is enabled on every 18 Kb block that is not instantiated in the design to save power. Power-gated 18 Kb blocks are not initialized during configuration and retain their house keeping value of zero. A valid bitstream is required for configuration and readback. Blank bitstreams are not allowed. The access to uninstantiated block RAM is prevented by disabling the internal operation.

Block RAM Library Primitives

The following table lists the block RAM primitives.

Table 6: **Block RAM**

Primitive	Description
RAMB36E5 RAMB18E5	• Versal devices feature distributed RAM and high-speed SelectIO™memory interfaces, as well as numerous 36 Kb block RAMs. • Each 36 Kb block RAM contains two independently controlled 18 Kb RAMs and can be used either way. • Block RAMs are placed in columns, and the total number of block RAM depends on the size of the Versal device. • Embedded dual- or single-port RAM modules, ROM modules, and data width converters are implemented with Xilinx CORE Generator™ block memory modules.

The block RAM library primitives RAMB18E5 and RAMB36E5 are the basic building blocks for all block RAM configurations. Other block RAM primitives and macros are based on these primitives. Some block RAM attributes can only be configured using one of these primitives (for example, pipeline register, cascade).

The input and output data buses are represented by two buses for 9-bit width (8 + 1), 18-bit width (16 + 2), and 36-bit width (32 + 4) configurations. The ninth bit associated with each byte can store parity/error correction bits or serve as additional data bits. No specific function is performed on the ninth bit. The separate bus for parity bits facilitates some designs. However, other designs safely use a 9-bit, 18-bit, or 36-bit bus by merging the regular data bus with the parity bus. Read/write and storage operations are identical for all bits, including the parity bits.

The following figure illustrates all the I/O ports of the 36 Kb true dual-port block RAM primitive (RAMB36). The following table lists these primitives.

***Note*:** ECC pins are not shown in the figure. For more information, see Built-in Error Correction.

Figure 12: **Block RAM Port Signals (RAMB36E5)**

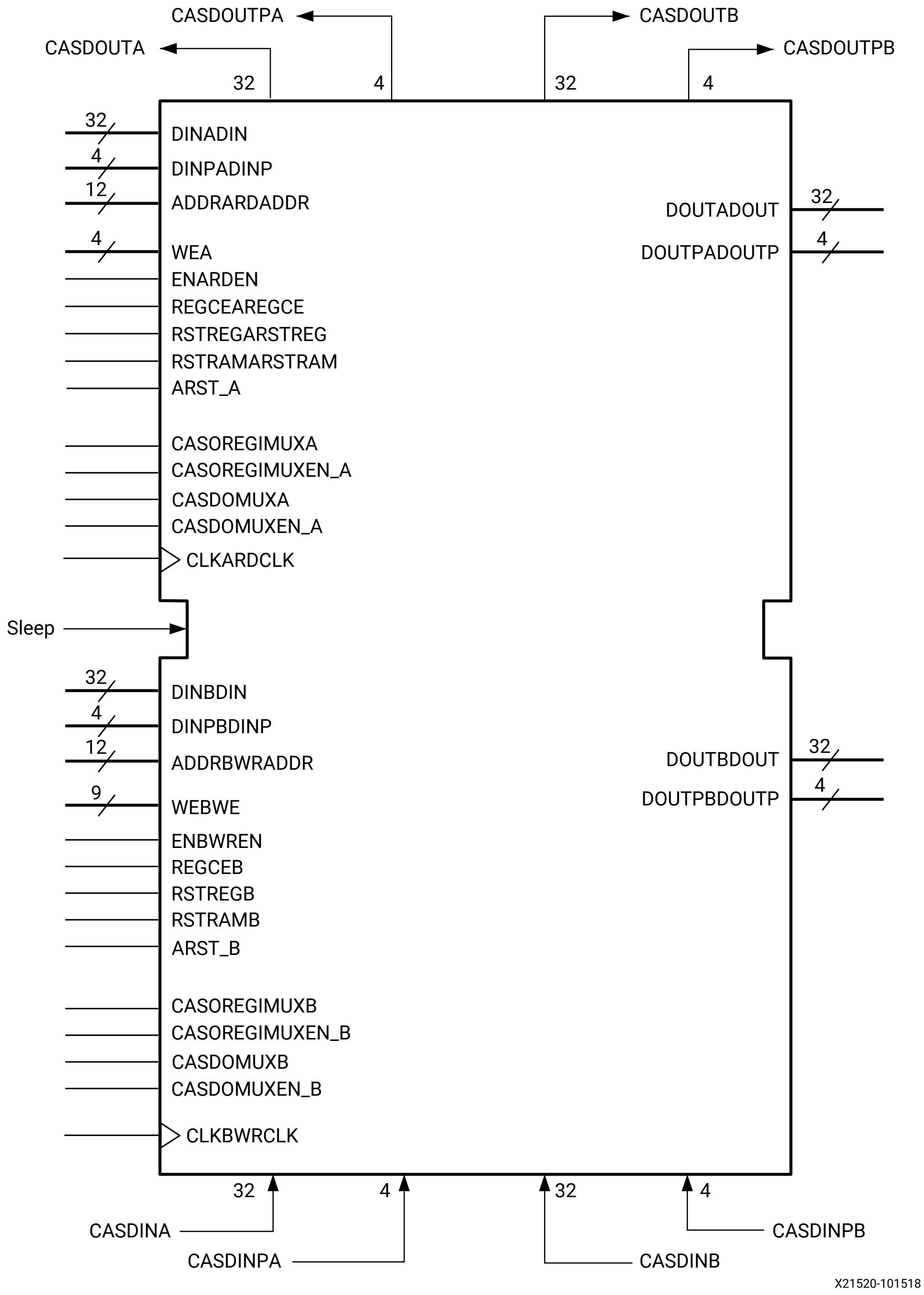


Table 7: **Block RAM Primitives**

Primitive	Description
RAMB36E5	• When used as TDP memory, RAMB36E5 supports port widths of x9, x18, and x36. • When used as SDP memory, the read or write port width is x64 or x72. Alternate port widths are x9, x18, x36, and x72. In ECC mode, RAMB36E5 supports 64-bit ECC encoding and decoding.
RAMB18E5	• When used as TDP memory, RAMB18E5 supports port widths of x9 and x18. • When used as SDP memory, the read or write port width is x32 or x36. Alternate port widths are x9, x18, and x36.

The following table shows the port names and descriptions of the primitives outlined in the previous table. The ECC ports are described in Built-in Error Correction.

Table 8: **RAMB36E5 and RAMB18E5 Port Names and Descriptions**

Port Name	Description
DINADIN[31:0]	Port A data inputs addressed by ADDRARDADDR. See Table 13 for SDP usage port name mapping.
DINPADINP[3:0]	Port A data parity inputs addressed by ADDRARDADDR. See Table 13 for SDP usage port name mapping.
DINBDIN[31:0]	Port B data inputs addressed by ADDRBWRADDR. See Table 13 for SDP usage port name mapping.
DINPBDINP[3:0]	Port A data parity inputs addressed by ADDRBWRADDR. See Table 13 for SDP usage port name mapping.
ADDRARDADDR[11:0]	Port A address input bus. When used as SDP memory, this is the RDADDR bus.
ADDRARDADDR[10:0]	(RAMB18E5 ONLY) Port A address input bus. When used as SDP memory, this is the RDADDR bus.
ADDRBWRADDR[11:0]	Port B address input bus. When used as SDP memory, this is the WRADDR bus.
ADDRBWRADDR[10:0]	(RAMB18E5 ONLY) Port B address input bus. When used as SDP memory, this is the WRADDR bus.
ARST_A	Asynchronous reset that resets the output register for port A to all zeros.
ARST_B	Asynchronous reset that resets the output register for port B to all zeros.
WEA[3:0]	Port A byte-wide write enable. When used as SDP memory, this port is not used.
WEBWE[8:0]	Port B byte-wide write enable. In SDP mode, this is the byte-wide write enable.
ENARDEN	Port A enable. When used as SDP memory, this is RDEN.
ENBWREN	Port B enable. When used as SDP memory, this is WREN.
RSTREGARSTREG	Synchronous output register set/reset as initialized by SRVAL_A (DOA_REG = 1). RSTREG_PRIORITY_A determines the priority over REGCE. When used as SDP memory, this is RSTREG.
RSTREGB	Synchronous output register set/reset as initialized by SRVAL_B (DOA_REG = 1). RSTREG_PRIORITY_B determines the priority over REGCE.
RSTRAMARSTRAM	Synchronous output latch set/reset as initialized by SRVAL_A (DOB_REG = 0). When used as SDP memory, this is RSTRAM.
RSTRAMB	Synchronous output latch set/reset as initialized by SRVAL_B (DOB_REG = 0).

Table 8: **RAMB36E5 and RAMB18E5 Port Names and Descriptions** *(cont'd)*

Port Name	Description
CLKARDCLK	Port A clock input. When used as SDP memory, this is RDCLK.
CLKBWRCLK	Port B clock input. When used as SDP memory, this is WRCLK.
REGCEAREGCE	Port A output register clock enable (DOA_REG = 1). When used as SDP memory, this is REGCE.
REGCEB	Port B output register clock enable (DOB_REG = 1).
CASDINA[31:0]	Port A cascade data input connected to data output of lower block RAM. For RAMB18E5: CASDINA[15:0].
CASDINPA[3:0]	Port A cascade parity data input connected to parity data output of lower block RAM. For RAMB18E5: CASDINPA[1:0].
CASDINB[31:0]	Port B cascade data input connected to data output of lower block RAM. For RAMB18E5: CASDINB[15:0].
CASDINPB[3:0]	Port B cascade parity data input connected to parity data output of lower block RAM. For RAMB18E5: CASDINPB[1:0].
CASDOUTA[31:0]	Port A cascade data output connected to CASDINA[31:0] of upper block RAM. For RAMB18E5: CASDOUTA[15:0].
CASDOUTPA[3:0]	Port A cascade parity data output connected to CASDINPA[3:0] of upper block RAM. For RAMB18E5: CASDOUTPA[1:0].
CASDOUTB[31:0]	Port B cascade data output connected to CASDINB[31:0] of upper block RAM. For RAMB18E5: CASDOUTB[15:0].
CASDOUTPB[3:0]	Port B cascade parity data output connected to CASDINPB[3:0] of upper block RAM. For RAMB18E5: CASDOUTPB[1:0].
CASDOMUXA	Selects input to control the data cascade output multiplexer for port A.
CASDOMUXEN_A	Enables control for the CASDOMUXA register.
CASDOMUXB	Selects input to control the data cascade output multiplexer for port B.
CASDOMUXEN_B	Enables control for the CASDOMUXB register. When used as SDP memory, this port is not used.
CASOREGIMUXA	Selects input to control the cascade multiplexer before the output register for Port A.
CASOREGIMUXEN_A	Enables control for the CASOREGIMUXA register.
CASOREGIMUXB	Selects input to control the cascade multiplexer before the output register for Port B. When used as SDP memory, this port is not used.
CASOREGIMUXEN_B	Enables control for the CASOREGIMUXB register. When used as SDP memory, this port is not used.
DOUTADOUT[31:0]	Port A data output bus addressed by ADDRARDADDR. See Table 13 for SDP usage port name mapping. RAMB18E5: DOUTADOUT[15:0].
DOUTPADOUTP[3:0]	Port A parity output bus addressed by ADDRARDADDR. See Table 13 for SDP usage port name mapping. RAMB18E5: DOUTPADOUTP[1:0].
DOUTBDOUT[31:0]	Port B data output bus addressed by ADDRBWRADDR. See Table 13 for SDP usage port name mapping. RAMB18E5: DOUTBDOUT[15:0].
DOUTPBDOUTP[3:0]	Port B parity output bus addressed by ADDRBWRADDR. See Table 13 for SDP usage port name mapping. RAMB18E5: DOUTPBDOUTP[1:0].
SLEEP	Dynamic power gating.

Block RAM Port Signals

Each block RAM port operates independently of the other while accessing the same set of 36 Kbit memory cells.

Clock – CLKARDCLK and CLKBWRCLK

Each port is fully synchronous with independent clock pins. All port input pins have setup time referenced to the port CLK pin. The output data bus has a clock-to-out time referenced to the CLK pin. Clock polarity is configurable (rising edge by default). When used as SDP memory, the CLKA port is the RDCLK and the CLKB port is the WRCLK.

Enable – ENARDEN and ENBWREN

The enable pin affects the read, write, and set/reset functionality of the port. Ports with an inactive enable pin keep the output pins in the previous state and do not write data to the memory cells. Enable polarity is configurable (active-High by default). When used as SDP memory, the ENA port is the RDEN and the ENB port is the WREN.

Byte-Wide Write Enable – WEA and WEBWE

To write the content of the data input bus into the addressed memory location, both EN and WE must be active within a setup time before the active clock edge. The output latches are loaded or not loaded according to the write configuration (WRITE_FIRST, READ_FIRST, NO_CHANGE). When WE is inactive and EN is active, a read operation occurs, and the contents of the memory cells referenced by the address bus appear on the data out bus, regardless of the write mode attribute. Write enable polarity is not configurable (active-High). When used as SDP memory, the WEBWE[8:0] port is the byte-write enable. When used as TDP memory, the WEA[3:0] and WEB[3:0] are byte-write enables for port A and port B, respectively.

Related Information

Byte-Wide Write Enable

Register Enable – REGCEAREGCE and REGCEB

The register enable pin (REGCE) controls the optional output register. When the block RAM is in register mode, REGCE = 1 registers the output into a register at a clock edge. The polarity of REGCE is not configurable (active-High). When used as SDP memory, the REGCEA port is the REGCE.

Set/Reset

RSTREGARSTREG, RSTREGB, RSTRAMARSTRAM, RSTRAMB, and ARST

In latch mode, the RSTRAM pin synchronously forces the data output latches to contain the value SRVAL. When the optional output registers are enabled (DO_REG = 1), the RSTREG signal synchronously forces the data output registers containing the SRVAL value. The priority of RSTREG over REGCE is determined using the RSTREG_PRIORITY attribute. The data output latches or output registers are synchronously asserted to 0 or 1, including the parity bit. Each port has an independent SRVAL[A|B] attribute of 36 bits. This operation does not affect RAM memory cells and does not disturb write operations on the other port. The polarity for both signals is configurable (active-High by default). When used as SDP memory, the RSTREGA port is the RSTREG, and the RSTRAMA port is the RSTRAM.

ARST_A and ARST_B are asynchronous resets for port A and port B which reset port A and port B outputs to all zeros, respectively. If asynchronous mode is enabled, the output of all pipeline stages is asynchronously set to 0 when ARST is asserted. This is regardless of the block RAM being enabled or the setting of the SRVAL value. Latch mode, register mode, and eccpipe mode are affected by the asynchronous reset. Toggling the synchronous reset pins while in the asynchronous reset mode has no impact on the outputs. All block RAMs in a cascade chain must be driven by the same synchronized input signal.

***Note*:** The user design should synchronize the falling edge to meet recovery and removal timing. The block RAM does not have synchronizers for the asynchronous reset inputs.

Address Bus – ADDRARDADDR and ADDRBWRADDR

The address bus selects the memory cells for read or write. When used as SDP memory, the ADDRA port is the RDADDR, and the ADDRB port is the WRADDR. The data bit width of the port determines the required address bus width for a single RAMB18E5 or RAMB36E5, as shown in the following tables.

Table 9: **Port Aspect Ratio for RAMB18E5 (When Used as TDP Memory)**

Port Data Width	Port Address Width	Depth	ADDR Bus	DIN Bus DOUT Bus	DINP Bus DOUTP Bus
9	11	2,048	[10:0]	[7:0]	[0]
18	10	1,024	[10:1]	[15:0]	[1:0]

Table 10: **Port Aspect Ratio for RAMB18E5 (When Used as SDP Memory)**

Port Data Width [1]	Alternate Port Width	Port Address Width	Depth	ADDR Bus	DIN Bus DOUT Bus	DINP Bus DOUTP Bus
36	9	11	2,048	[10:0]	[7:0]	[0]
36	18	10	1,024	[10:1]	[15:0]	[1:0]
36	36	9	512	[10:2]	[31:0]	[3:0]

Notes:

1. Either the read or write port is a fixed width of x32 or x36.

Table 11: **Port Aspect Ratio for RAMB36E5 (When Used as TDP Memory)**

Port Data Width	Port Address Width	Depth	ADDR Bus	DIN Bus DOUT Bus	DINP Bus DOUTP Bus
9	12	4,096	[11:0]	[7:0]	[0]
18	11	2,048	[11:1]	[15:0]	[1:0]
36	10	1,024	[11:2]	[31:0]	[3:0]

Table 12: **Port Aspect Ratio for RAMB36E5 (When Used as SDP Memory)**

Port Data Width [1]	Alternate Port Width	Port Address Width	Depth	ADDR Bus	DIN Bus DOUT Bus	DINP Bus DOUTP Bus
72	9	12	4,096	[11:0]	[7:0]	[0]
72	18	11	2,048	[11:1]	[15:0]	[1:0]
72	36	10	1,024	[11:2]	[31:0]	[3:0]
72	72	9	512	[11:3]	[63:0]	[7:0]

Notes:

1. Either the read or write port is a fixed width of x64 or x72.

For block RAMs used as SDP memories, the port name mapping is listed in the following table. Figure 6 shows the SDP data flow.

Table 13: **Port Name Mapping for Block RAMs Used as SDP Memories**

RAMB18E5 Used as SDP Memory		RAMB36E5 Used as SDP Memory	
X36 Mode (Width = 36)	**X18 Mode (Width ≤ 18)**	**X72 Mode (Width = 72)**	**X36 Mode (Width ≤ 36)**
DIN[15:0] = DINADIN[15:0]	DIN[15:0] = DINBDIN[15:0]	DIN[31:0] = DINADIN[31:0]	DIN[31:0] = DINBDIN[31:0]
DINP[1:0] = DINPADIN[1:0]	DINP[1:0] = DINPBDINP[1:0]	DINP[3:0] = DINPADIN[3:0]	DINP[3:0] = DINPBDINP[3:0]
DIN[31:16] = DINBDIN[15:0]		DIN[63:32] = DINBDIN[31:0]	

Table 13: **Port Name Mapping for Block RAMs Used as SDP Memories** *(cont'd)*

RAMB18E5 Used as SDP Memory		RAMB36E5 Used as SDP Memory	
X36 Mode (Width = 36)	**X18 Mode (Width ≤ 18)**	**X72 Mode (Width = 72)**	**X36 Mode (Width ≤ 36)**
DINP[3:2] = DINPBDINP[1:0]		DINP[7:4] = DINPBDINP[3:0]	
DOUT[15:0] = DOUTADOUT[15:0]	DOUT[15:0] = DOUTADOUT[15:0]	DOUT[31:0] = DOUTADOUT[31:0]	DOUT[31:0] = DOUTADOUT[31:0]
DOUTP[1:0] = DOUTPADOUTP[1:0]	DOUTP[1:0] = DOUTPADOUTP[1:0]	DOUTP[3:0] = DOUTPADOUTP[3:0]	DOUTP[3:0] = DOUTPADOUTP[3:0]
DOUT[31:16] = DOUTBDOUT[15:0]		DOUT[63:32] = DOUTBDOUT[31:0]	
DOUTP[3:2] = DOUTPBDOUTP[1:0]		DOUTP[7:4] = DOUTPBDOUTP[3:0]	

Data-In Buses – DINADIN, DINPADINP, DINBDIN, and DINPBDINP

Data-in buses provide the new data value to be written into RAM. The regular data-in bus (DIN) plus the data-in parity bus (DINP), when available, have a total width equal to the port width. For example, the 36-bit port data width is represented by DIN[31:0] and DINP[3:0], as shown in Table 9 through Table 12. See Table 13 for port name mapping for block RAMs used as SDP memories.

Data-Out Buses – DOUTADOUT, DOUTPADOUTP, DOUTBDOUT, and DOUTPBDOUTP

Data-out buses reflect the contents of memory cells referenced by the address bus at the last active clock edge during a read operation. During a write operation (WRITE_FIRST or READ_FIRST configuration), the data-out buses reflect either the data being written or the stored value before write. During a write operation in NO_CHANGE mode, data-out buses are not changed. The regular data-out bus (DOUT) plus the parity data-out bus (DOUTP) (when available) have a total width equal to the port width, as shown in Table 9 through Table 12. See Table 13 for port name mapping for block RAMs used as SDP memories.

CASDINA

This is the data input cascade for port A from the block RAM below.

Related Information

Cascadable Block RAM

CASDINB

This is the data input cascade for port B from the block RAM below.

CASDINPA

This is the parity input cascade for port A from the block RAM below.

CASDINPB

This is the parity input cascade for port B from the block RAM below.

CASDOUTA

This is the data output cascade for port A to the block RAM above.

CASDOUTB

This is the data output cascade for port B to the block RAM above.

CASDOUTPA

This is the parity output cascade for port A to the block RAM above.

CASDOUTPB

This is the parity output cascade for port B to the block RAM above.

Cascade Selection – CASOREGIMUX

This is the D input to the register that drives the multiplexer select line to select between regular data from the block RAM output or the cascade input (CASDIN) when the block RAM is in cascade mode. This multiplexer is before the optional output register and adds a pipeline stage in cascade mode. When the block RAM is not used in cascade mode, block RAM data is always selected.

Cascade Selection – CASOREGIMUXEN

This is the enable control input to the register that drives the multiplexer select line to select between regular data from the block RAM output or the cascade input (CASDIN).

Cascade Selection – CASDOMUX

This is the register D input that drives the output multiplexer select line to select between regular data from the block RAM output or the cascade input (CASDIN) when the block RAM is in cascade mode. This multiplexer is after the optional output register. When the block RAM is not used in cascade mode, block RAM data is always selected.

Cascade Selection – CASDOMUXEN

This is the enable control input to the register that drives the select line to the cascade output multiplexer of the block RAM outputs in cascade mode.

SLEEP

The SLEEP pin provides a dynamic power gating capability for periods when the block RAM is not actively used for an extended period of time. While SLEEP is active (High) the EN pins on both ports must be held Low. The data content of the memory is preserved during this mode. There is a wake-up time requirement of two clock cycles regardless of the SLEEP_ASYNC mode setting. Any block RAM access prior to the wake-up time requirement is not guaranteed and might cause memory content corruption. The attribute SLEEP_ASYNC determines the behavior of this pin with respect to the clocks.

Related Information

Block RAM Attributes

Inverting Control Pins

For each port, the eight control pins (CLK, EN, ARST, RSTREG, and RSTRAM) each have an individual inversion option. EN, RSTREG, and RSTRAM control signals can be configured as active-High or Low, and the clock can be active on a rising or falling edge (active-High on rising edge by default), without requiring other logic resources.

RAMB18/36 Unused Inputs

Unused input pins require a certain defined constant input value for the block RAM to function properly. If left unconnected (Verilog), the Vivado tools automatically tie them to the appropriate constant value. However, if the inputs are connected to a constant in the design (VHDL), then the values listed in the following table are required.

The unused inputs are shown here.

Table 14: **RAMB18/36 Unused Inputs**

RAMB18/36	Constant	Comments
CLKARDCLK	0	
CLKBRDCLK	0	
CLKAWRCLK	0	
CLKBWRCLK	0	
ENARDEN	0	
ENBWREN	0	
REGCEAREGCE	1	Xilinx recommends setting to 0 when DOA_REG = 0 for power saving
REGCEB	1	Xilinx recommends setting to 0 when DOB_REG = 0 for power saving
REGCLKARDRCLK	0	
REGCLKB	0	
RSTREGARSTREG	0	
RSTREGB	0	
RSTRAMARSTRAM	0	
RSTRAMB	0	
RSTRAMARSTRAM	0	
RSTRAMB	0	
SLEEP	0	
WEA<3:0>	1	TDP: When not using port A for write, (WRITE_WIDTH_A = 0), WEA<0> must be connected to 0
WEBWE<8:0>	1	TDP: When not using port B for write (WRITE_WIDTH_B=0), WEB<0> must be connected to 0
CASDOMUXA	0	
CASDOMUXB	0	
CASOREGIMUXA	0	
CASOREGIMUXB	0	
CASDOMUXEN_A	1	
CASDOMUXEN_B	1	
CASOREGIMUXEN_A	1	
CASOREGIMUXEN_B	1	
INJECTSBITERR	0	
INJECTDBITERR	0	

Block RAM Address Mapping

Each port accesses the same set of 18,432 or 36,864 memory cells using an addressing scheme dependent on whether it is a RAMB18E5 or RAMB36E5. The physical RAM locations addressed for a particular width are determined using these formulae (of interest only when the two ports use different aspect ratios):

END = ((ADDR + 1) × Width) – 1

START = ADDR × Width

The following table shows low-order address mapping for each port width.

Table 15: **Port Address Mapping to Physical Memory Word Lines**

Port Width	Parity Locations				Data Locations			
8 + 1	3	2	1	0	Byte3	Byte2	Byte1	Byte0
16 + 2	1		0		Half-Word1		Half-Word0	
32 + 4	0				Word0			

Block RAM Attributes

The following table lists the RAMB18E5 and RAMB36E5 attributes. All attribute code examples are discussed in Block RAM Initialization in VHDL or Verilog Code. Further information on using these attributes is available in Additional RAMB18E5 and RAMB36E5 Primitive Design Considerations.

Table 16: **RAMB18E5 and RAMB36E5 Attributes**

Attributes	Values	Default	Type	Description
BWE_MODE_B	PARITY_INTERLEAVED, PARITY_INDEPENDENT	PARITY_INTERLEAVED	String	In PARITY_INTERLEAVED, WEBWE[8] should be connected to 1.
CASCADE_ORDER_A	FIRST, MIDDLE, LAST, NONE	NONE	String	Specifies the order of the cascaded block RAMs from the bottom to the top of the chain for port A.
CASCADE_ORDER_B	FIRST, MIDDLE, LAST, NONE	NONE	String	Specifies the order of the cascaded block RAMs from the bottom to the top of the chain for port B.
CLOCK_DOMAINS	INDEPENDENT, COMMON	INDEPENDENT	String	Either independent clocks connected to port A and B or a single, common clock connected to port A and B.
DOA_REG	0, 1	1	Decimal	A value of 1 enables the optional output registers of the RAM port A. Applies to all port A outputs in both TDP and SDP memory usage.

Table 16: **RAMB18E5 and RAMB36E5 Attributes** *(cont'd)*

Attributes	Values	Default	Type	Description
DOB_REG	0, 1	1	Decimal	A value of 1 enables the optional output registers of the RAM port B. Applies to all port B outputs in both TDP and SDP memory usage.
RAMB18E5: INIT_00 to INIT_3F RAMB36E5: INIT_00 to INIT_7F	A 256-bit hex value	All 0	Hex	Initializes the data content of the block RAM.
RAMB18E5: INITP_00 to INITP_07 RAMB36E5: INITP_00 to INITP_0F	A 256-bit hex value	All 0	Hex	Initializes the parity content of the block RAM.
READ_WIDTH_A	RAMB18E5: 0, 9, 18, 36 (SDP usage) RAMB36E5: 0, 9, 18, 36, 72 (SDP usage)	RAMB18E5: 0 RAMB36E5: 0	Decimal	Specifies the data width for read port A, including parity bits. In SDP mode, this is the read width including parity bits.
READ_WIDTH_B	RAMB18E5: 0, 9, 18 RAMB36E5: 0, 9, 18, 36	RAMB18E5: 0 RAMB36E5: 0	Decimal	Specifies the data width for read port B including parity bits. Not used for SDP memory usage.
RSTREG_PRIORITY_A	RSTREG, REGCE	RSTREG	String	Selects the priority of RESET or CE for the optional output registers. Applies to all port A outputs in both TDP and SDP memory usage.
RSTREG_PRIORITY_B	RSTREG, REGCE	RSTREG	String	Selects the priority of RESET or CE for the optional output registers. Applies to all port B outputs in both TDP and SDP memory usage.
RST_MODE_A	SYNC, ASYNC	SYNC	String	Determines whether RST_A is synchronous or asynchronous input.
RST_MODE_B	SYNC, ASYNC	SYNC	String	Determines whether RST_B is synchronous or asynchronous input.
SLEEP_ASYNC	FALSE, TRUE	FALSE	String	Determines if the SLEEP pin is synchronous or asynchronous to the clock.
SRVAL_A	RAMB18E5: 18-bit hex value RAMB36E5: 36-bit hex value	RAMB18E5: `18'h00000000` RAMB36E5: `36'h0000000000000000`	Hex	Specifies the initialization value of the output latches or register when the synchronous reset (RSTREG) is asserted. Applies to all port A outputs in both TDP and SDP memory usage.

Table 16: **RAMB18E5 and RAMB36E5 Attributes** *(cont'd)*

Attributes	Values	Default	Type	Description
SRVAL_B	RAMB18E5: 18-bit hex value RAMB36E5: 36-bit hex value	RAMB18E5: `18'h00000000` RAMB36E5: `36'h0000000000000000`	Hex	Specifies the initialization value of the output latches or register when the synchronous reset (RSTREG) is asserted. Applies to all port B outputs in both TDP and SDP memory usage.
WRITE_MODE_A[1]	WRITE_FIRST, NO_CHANGE, READ_FIRST	NO_CHANGE	String	Specifies output behavior of write port A. See Write Modes.
WRITE_MODE_B[1]	WRITE_FIRST, NO_CHANGE, READ_FIRST	NO_CHANGE	String	Specifies output behavior of write port B. See Write Modes.
WRITE_WIDTH_A	RAMB18E5: 0, 9, 18 RAMB36E5: 0, 9, 18, 36	RAMB18E5: 0 RAMB36E5: 0	Decimal	Specifies the data width for write port A, including parity bits. When used as SDP memory, this attribute is not valid.
WRITE_WIDTH_B	RAMB18E5: 0, 9, 18, 36 (SDP usage) RAMB36E5: 0, 9, 18, 36, 72 (SDP usage)	RAMB18E5: 0 RAMB36E5: 0	Decimal	Specifies the data width for write Port B, including parity bits. In SDP mode, this is the write width including parity bits.

Notes:

1. In SDP mode, the WRITE_MODE_A and WRITE_MODE_B must have the same value.

Data Cascading – CASCADE_ORDER

Specifies the order of the cascaded block RAM. The first block RAM is at the bottom in the cascade chain, the last one is on the top of the cascade, and the middle ones are the block RAM(s) in between bottom and top. This applies to ports A and/or B.

Clocking – CLOCK_DOMAINS

This attribute defines if the clocks to ports A and B are independent/asynchronous or common/synchronous. Clocks driven by the same clock source (CLKA and CLKB are connected together) are common. All other CLKA and CLKB connections are independent.

Content Initialization – INIT_xx

The memory content can be initialized or cleared in the configuration bitstream. A standard, valid bitstream is required for block RAM initialization or readback due to the power gating feature. For more details on initialization and readback of uninstantiated (power gated) block RAM, see Power Gating of Unused Block RAMs.

INIT_xx attributes define the initial memory contents. By default, block RAM is initialized with all zeros during the device configuration sequence. The 64 initialization attributes from INIT_00 through INIT_3F for the RAMB18E5, and the 128 initialization attributes from INIT_00 through INIT_7F for the RAMB36E5 represent the regular memory contents. Each INIT_xx is a 64-digit hex-encoded bit vector. The memory contents can be partially initialized and are automatically completed with zeros.

The following formula is used to determine the bit positions for each INIT_xx attribute. Given yy = conversion hex-encoded to decimal (xx), INIT_xx corresponds to the memory cells as follows:

- from [(yy + 1) × 256] – 1
- to (yy) × 256

For example, for the attribute INIT_1F, the conversion is:

- yy = conversion hex-encoded to decimal (xx) "`1F`" = 31
- from [(31 + 1) × 256] – 1 = 8,191
- to 31 × 256 = 7,936

More examples are given in the following table.

Table 17: **Block RAM Initialization Attributes**

Attribute	Memory Location	
	From	To
INIT_00	255	0
INIT_01	511	256
INIT_02	767	512
...	...	...
INIT_0E	3839	3584
INIT_0F	4095	3840
INIT_10	4351	4096
...	...	...
INIT_1F	8191	7936
INIT_20	8447	8192
...	...	...
INIT_2F	12287	12032
INIT_30	12543	12288
...	...	...
INIT_3F	16383	16128
...	...	...
INIT_7F	32767	32512

Content Initialization – INITP_xx

INITP_xx attributes define the initial contents of the memory cells corresponding to DINP/ DOUTP buses (parity bits). By default, these memory cells are also initialized to all zeros. The initialization attributes represent the memory contents of the parity bits. The eight initialization attributes are INITP_00 through INITP_07 for the RAMB18E5. The 16 initialization attributes are INITP_00 through INITP_0F for the RAMB36E5. Each INITP_xx is a 64-digit hex-encoded bit vector with a regular INIT_xx attribute behavior. The same formula can be used to calculate the bit positions initialized by a particular INITP_xx attribute.

Read Width – READ_WIDTH_[A|B]

This attribute determines the A/B read port width of the block RAM. The valid values are: 0 (default), 9, 18, 36, and 72 for the RAMB36E5 when used as SDP memory.

Reset or CE Priority – RSTREG_PRIORITY_[A|B]

This attribute determines the priority of RSTREG or REGCE while asserting RSTREG when DO_REG = 1. Valid values are RSTREG or REGCE. When RSTREG has priority, the RSTREG input resets the optional output register, regardless of the state of REGCE. When REGCE has priority, the RSTREG input resets the optional output register only when REGCE = 1.

Asynchronous/Synchronous Reset Mode Setting - RST_MODE_[A/B]

This attribute determines if the user reset signal is synchronous or asynchronous. By default this is set to synchronous, which means the user can reset the block RAM by using RSTRAM or RSTREG (if output registers are used) to a specified SRVAL. If this attribute is set to asynchronous, ARST_[A|B] resets all pipe stages of the block RAM to 0. The value of the RSTRAM and RSTREG inputs are ignored and not propagated to the block RAM circuit. The SRVAL setting is also ignored.

Power Saving – SLEEP_ASYNC

This attribute determines if the SLEEP pin is to be used in synchronous or asynchronous mode. Synchronous mode (SLEEP_ASYNC = FALSE) should be used when either both clocks are identical or have a fixed phase relationship. In this mode, ENA and ENB must be deasserted (disabled) in the clock cycle prior to asserting SLEEP. The assertion and deassertion of SLEEP must meet the setup and hold times with respect to both CLKA and CLKB. ENA and ENB must only be asserted again after the block RAM returns from its sleep mode after two clock cycles.

Asynchronous mode (SLEEP_ASYNC = TRUE) should be used when both clocks are truly independent (asynchronous to each other). In this mode, ENA and ENB must be deasserted (disabled) in the clock cycle for the slowest clock prior to asserting SLEEP. SLEEP can then be asserted with the next clock cycle of the same clock. The deassertion of SLEEP causes the block RAM to activate (wake up) up after two clock cycles. Only after the memory wakeup can ENA and ENB be asserted again.

Output Latches/Registers Synchronous Set/Reset (SRVAL_[A|B])

The SRVAL (single-port) or SRVAL_A and SRVAL_B (dual-port) attributes define output latch values when the RSTRAM/RSTREG input is asserted. The width of the SRVAL (SRVAL_A and SRVAL_B) attribute is the port width, as shown in the table. These attributes are hex-encoded bit vectors and the default value is 0. This attribute sets the value of the output register when the optional output register attribute is set. When the register is not used, the latch gets set to the SRVAL instead. The following tables show how the SRVAL and INIT bit locations map to the DOUT outputs for the block RAM primitives and the SDP macro.

Table 18: **RAMB18E5 and RAMB36E5, SRVAL Mapping for Port A and Port B**

Port Width	SRVAL_(A/B) Full Width	SRVAL_(A/B) Mapping to DOUT		SRVAL_(A/B) Mapping to DOUTP	
		DOUTADOUT/ DOUTBDOUT	SRVAL_(A/B)	DOUTP(A/B)/ DOUTP	SRVAL_(A/B)
9	[8:0]	[7:0]	[7:0]	[0]	[8]
18	[17:0]	[15:0]	[15:0]	[1:0]	[17:16]
36 (only for RAMB36E5)	[35:0]	[31:0]	[31:0]	[3:0]	[35:32]

Table 19: **SDP Mapping for RAMB18E5 and RAMB36E5**

Port Width	SRVAL Full Width	SRVAL Mapping to DOUT		SRVAL Mapping to DOUTP	
		DOUT	SRVAL	DOUTP	SRVAL
36-bit wide RAMB18E5	[35:0]	[31:0]	[33:18]/[15:0]	[3:0]	[35:34]/[17:16]
72-bit wide RAMB36E5	[71:0]	[63:0]	[67:36]/[31:0]	[7:0]	[71:68]/[35:32]

Optional Output Register On/Off Switch – DOUT[A|B]_REG

This attribute sets the optional pipeline registers at the A/B output of the block RAM improving the clock-to-out timing. If turned on, this adds an extra cycle of read latency. When turned off, the block RAM data is read in the same clock cycle, however with a slower clock-to-out. The valid values are 0 (default) or 1.

Write Width – WRITE_WIDTH_[A|B]

This attribute determines the A/B write port width of the block RAM. The valid values are: 0 (default), 9, 18, 36, and 72 for the RAMB36E5 when used as SDP memory.

Write Mode – WRITE_MODE_[A|B]

This attribute determines the write mode of the A/B input ports. The possible values are WRITE_FIRST (default), READ_FIRST, and NO_CHANGE.

Related Information
Write Modes

SIM_COLLISION_CHECK

This attribute sets the level of collision checking and behavior in the simulation model. Possible values are ALL (default), GENERATE_X_ONLY, NONE, and WARNING_ONLY.

INIT_FILE

This attribute points to an optional RAM initialization file (initial content). The values are NONE (default) or a STRING (the file name). For the file format, see the software documentation.

Block RAM Initialization in VHDL or Verilog Code

Block RAM attributes and content can be initialized in VHDL or Verilog code for both synthesis and simulation by using generic maps (VHDL) or defparams (Verilog) within the instantiated component. Modifying the values of the generic map or defparam affects both the simulation behavior and the implemented synthesis results. Inferred block RAM can be initialized as well. The Vivado Design Suite templates include the code to instantiate the RAMB primitives, as well as Xilinx Parameterized Macros (XPMs).

Additional RAMB18E5 and RAMB36E5 Primitive Design Considerations

The following is a checklist of block RAM design considerations.

Optional Output Registers

Optional output registers can be used at either or both A|B output ports of RAMB18E5 and RAMB36E5. The choice is made using the DO[A|B]_REG attribute. The two independent clock enable pins are REGCE[A|B]. When using the optional output registers at port [A|B], assertion of the synchronous set/reset (RSTREG and RSTRAM) pins of ports [A|B] causes the value specified by the attribute SRVAL to be registered at the output. Figure 13 shows an optional output register.

Using the output register is strongly recommended to enable a higher clock rate. However, it adds a clock cycle latency of one.

Output Register Reset

The block RAM synchronous output registers (optional) are set or reset (SRVAL) with RSTREG when DO_REG = 1. The RSTREG_PRIORITY attribute determines if RSTREG has priority over REGCE. The block RAM register mode RSTREG requires REGCE = 1 to reset the output DO register value if the RSTREG_PRIORITY is set to REGCE.

The synchronous output latches are set or reset (SRVAL) with RSTRAM when DO_REG is 0 or 1. The block RAM latch mode RSTRAM requires the block RAM enable, EN = 1, to reset the output DO latch value.

Independent Read and Write Port Width

***Note*:** To specify the port widths using the dual-port mode of the block RAM, designers must use the READ_WIDTH_[A|B] and WRITE_WIDTH_[A|B] attributes.

These rules should be considered:

- Designing a single-port block RAM requires the port pair widths of one write and one read to be set (for example, READ_WIDTH_A and WRITE_WIDTH_A).
- Designing a dual-port block RAM requires all port widths to be set.
- In simple dual-port mode, one side of the ports is fixed while the other side can have a variable width. The RAMB18E5 has a data port width of up to 36, while the RAMB36E5 has a data port width of up to 72. When using the block RAM as read-only memory, only the READ_WIDTH_A/B is used.

RAMB18E5 and RAMB36E5 Port Mapping Design Rules

The block RAMs are configurable to various port widths and sizes. Depending on the configuration, some data pins and address pins are not used. Table 9 through Table 12 show the pins used in various configurations. In addition to the information in these tables, these rules are useful to determine the RAMB port connections:

- When using RAMB36E5, if the DIN[A|B] pins are less than 32 bits wide, concatenate (32 - DIN_BIT_WIDTH) logic zeros to the front of DIN[A|B].
- If the DINP[A|B] pins are less than 4 bits wide, concatenate (4 - DINP_BIT_WIDTH) logic zeros to the front of DINP[A|B]. DINP[A|B] can be left unconnected when not in use.
- DOUT[A|B] pins must be 32 bits wide. However, valid data are only found on pins DOUT_BIT_WIDTH – 1 down to 0.
- DOUTP[A|B] pins must be 4 bits wide. However, valid data are only found on pins DOUTP_BIT_WIDTH – 1 down to 0. DOUTP[A|B] can be left unconnected when not in use.
- For the RAMB18E5, ADDR[A/B] is 11 bits wide and for the RAMB32E5, ADDR[A/B] is 12 bits wide. Address width is defined in Table 9.

Byte-Wide Write Enable

Consider these rules when using the byte-wide write enable feature:

- For RAMB36E5
- ◦ In x72 SDP mode, WEBWE[8:0] is used to connect the nine WE inputs for the write port. WEA[3:0] is not used. Depending on the BWE_MODE_B the parity bits are either interleaved with the data bits controlled by WEBWE[7:0] or controlled separately by WEBWE[8].

  - In x36 mode, WEA[3:0] is used to connect the four WE inputs for port A and WEBWE[3:0] is used to connect the four WE inputs for port B. WEBWE[7:4] is not used.
  - In x18 mode, WEA[1:0] is used to connect the two user WE inputs for port A and WEBWE[1:0] is used to connect the two WE inputs for port B. WEA[3:2] and WEBWE[7:2] are not used.
  - In x9 or smaller port width mode, WEA[0] is used to connect the single user WE input for port A and WEBWE[0] is used to connect the single WE input for port B. WEA[3:1] and WEBWE[7:1] are not used.

- For RAMB18E5
- - In x36 SDP mode, WEBWE[3:0] is used to connect the four WE inputs for the write port. WEA[1:0] is not used.
  - In x18 mode, WEA[1:0] is used to connect the two WE inputs for port A and WEBWE[1:0] is used to connect the two WE inputs for port B. WEBWE[3:2] is not used.
  - In x9 or smaller port width mode, WEA[0] is used to connect the single user WE input for port A and WEBWE[0] is used to connect the single WE input for port B. WEA[1] and WEBWE[3:1] are not used.

Related Information

Byte-Wide Write Enable – WEA and WEBWE

Minimum Clock Pulse and Address/Enable Setup/ Hold Time (Caution: Unstable Clocks)

IMPORTANT! *The clock minimum pulse width and setup/hold time of the block RAM address, block enable, and write enable pins must not be violated. Violating the clock minimum pulse width or these setup/hold times (even if write enable is Low) can corrupt the data contents of the block RAM. This most commonly occurs during an unstable clock (for example, when unplugging an external clock source) or when flip-flops driving block RAM control pins are asynchronously reset, such as a system-wide reset. To avoid this issue, design with synchronous resets only for both assertion and deassertion. When the clock is not stable, disable the clock buffer or the logic driving the block RAM control pins, or deassert the block RAM EN input.*

Block RAM Applications

Block RAM RSTREG in Register Mode

A block RAM RSTREG in register mode can be used to control the output register as a true pipeline register independent of the block RAM. As shown in the following figure, block RAMs can be read and written independent of register enable or set/reset. In register mode, RSTREG sets DOUT to the SRVAL and data can be read from the block RAM to DBRAM. Data at DBRAM can be clocked out (DOUT) on the next cycle. The timing diagrams in the following figures show different cases of the RSTREG operation.

Figure 13: **Block RAM RSTREG in Register Mode**

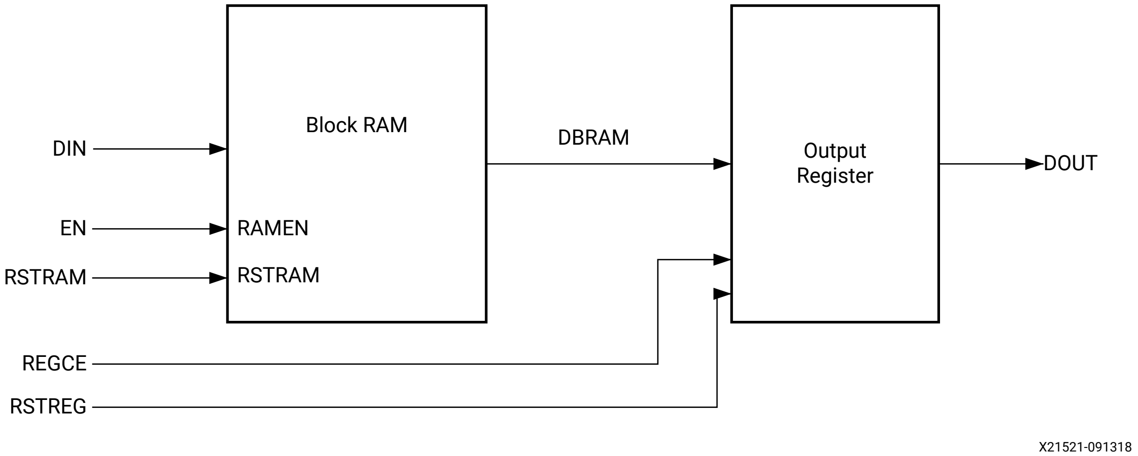


Figure 14: **Block RAM Reset Operation in RSTREG Mode**

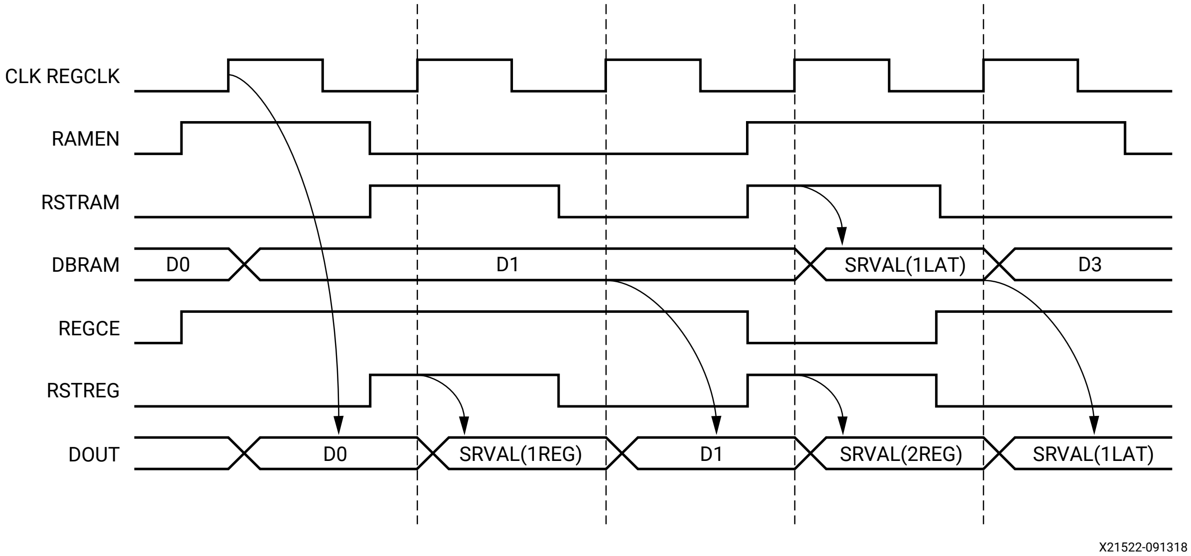


Figure 15: **Block RAM Reset Operation in REGCE Mode**

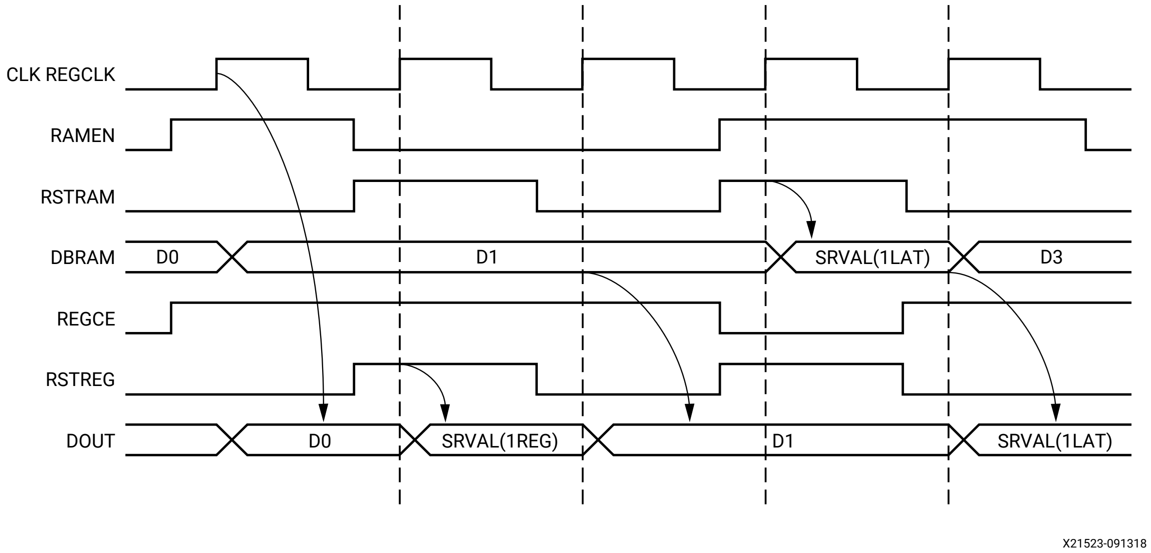
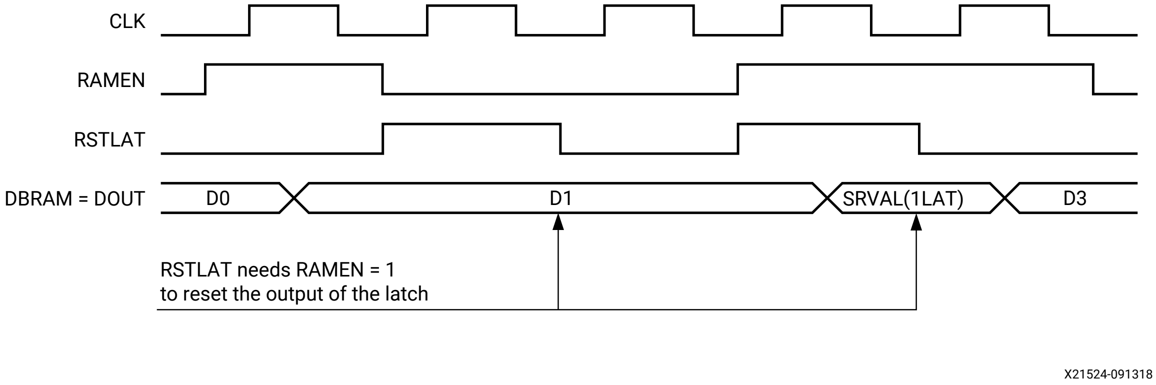


Figure 16: **Block RAM Reset Operation in Latch Mode**



Built-in Error Correction

The RAMB36E5 in simple dual-port mode can be configured as a single 512 x 64 RAM with built-in Hamming code error correction using the extra eight bits in the 72-bit wide RAM. This operation is transparent.

Eight protection bits (ECCPARITY) are generated during each write operation and stored with the 64-bit data into the memory. These ECCPARITY bits are used during each read operation to correct any single-bit error, or to detect (but not correct) any double-bit error. The ECCPARITY bits are written into the memory.

During each read operation, 72 bits of data (64 bits of data and 8 bits of parity) are read from the memory and fed into the ECC decoder. The ECC decoder generates two status outputs (SBITERR and DBITERR) that are used to indicate the three possible read results: No error, single-bit error corrected, and double-bit error detected. In the standard ECC mode, the read operation does not correct the error in the memory array, it only presents corrected data on DOUT. To improve F_{MAX}, optional registers controlled by the DO_REG attribute are available for data output (DOUT), SBITERR, and DBITERR. This is similar to the optional registers in the block RAM. For further F_{MAX} improvements, an additional ECC pipeline stage is available.

The ECC configuration option is available with a 36 Kb block RAM (RAMB36E5) in simple dual-port mode 72-bit width (64/8) (SDP). Both read and write width must be 72 bits. The RAMB36E5 has the capability to inject errors. The block RAM ECC also supports READ_FIRST, WRITE_FIRST, and NO_CHANGE modes in identical fashion to the SDP usage model.

ECC Modes

In the standard ECC mode (EN_ECC_READ = TRUE and EN_ECC_WRITE = TRUE), both encoder and decoder are enabled. During a write, 64-bit data and 8-bit ECC generated parity are stored in the array. The external parity bits are ignored. During a read, the 72-bit decoded data and parity (64-bit data and 8-bit parity) are read out, and the parity is checked against the data on DOUT.

The encoder and decoder can be accessed separately (independently) for external use in RAMB36E5 in simple dual-port mode in 72-bit mode. To use the encoder by itself, the data needs to be sent through the DIN port. To use the decoder by itself, the encoder is disabled, the data is written into the block RAM, and the corrected data and status bits are read out of the block RAM.

The decoder can be used in two ways:

- To use the decoder in standard ECC mode, set (EN_ECC_WRITE = TRUE and EN_ECC_READ = TRUE).
- To use the decoder-only mode, set (EN_ECC_WRITE = FALSE and EN_ECC_READ = TRUE). The DIN data along with a user-generated parity is presented on the eight DINP ports. The data is not encoded. The read operation reads back the data on DOUT and performs the decoder parity check on the data.

The encoder can be used in two ways:

- To use the encoder in standard ECC mode, set (EN_ECC_WRITE = TRUE and EN_ECC_READ = TRUE).
- To use the encoder-only mode, set (EN_ECC_WRITE = TRUE and EN_ECC_READ = FALSE). The DIN data is presented, and the ECC encoded value of the DIN data is stored in the parity bits for every write operation. The read operation reads those eight bits without performing the decode function.

The functionality of the block RAM when using the ECC mode is described as follows:

- The block RAM ports still have independent address, clocks, and enable inputs, but one port is a dedicated write port, and the other is a dedicated read port (simple dual-port).
- DOUT represents the read data after correction.
- DOUT stays valid until the next active read operation.
- Simultaneous decoding and encoding of different read/write addresses is allowed. However, simultaneous decoding and encoding of the same read/write address is not allowed.
- In ECC configuration, the block RAM can be in either READ_FIRST, WRITE_FIRST, and NO_CHANGE mode.

Versal devices have an ECC pipeline mode. This is in addition to the optional registers on the outputs. These registers effectively pipeline the decoder for further improvement in maximum performance (F_{MAX}) and clock-to-out in latch mode. If turned on, the latency increases by one clock cycle because while the current address is read from the block RAM, the previous address is being decoded. The ECC pipeline register has a user accessible ENABLE control but does not have a reset control. Asserting the block RAM reset pins RSTRAM and RSTREG has no impact on this register, and the previously registered data remains in the register.

Related Information

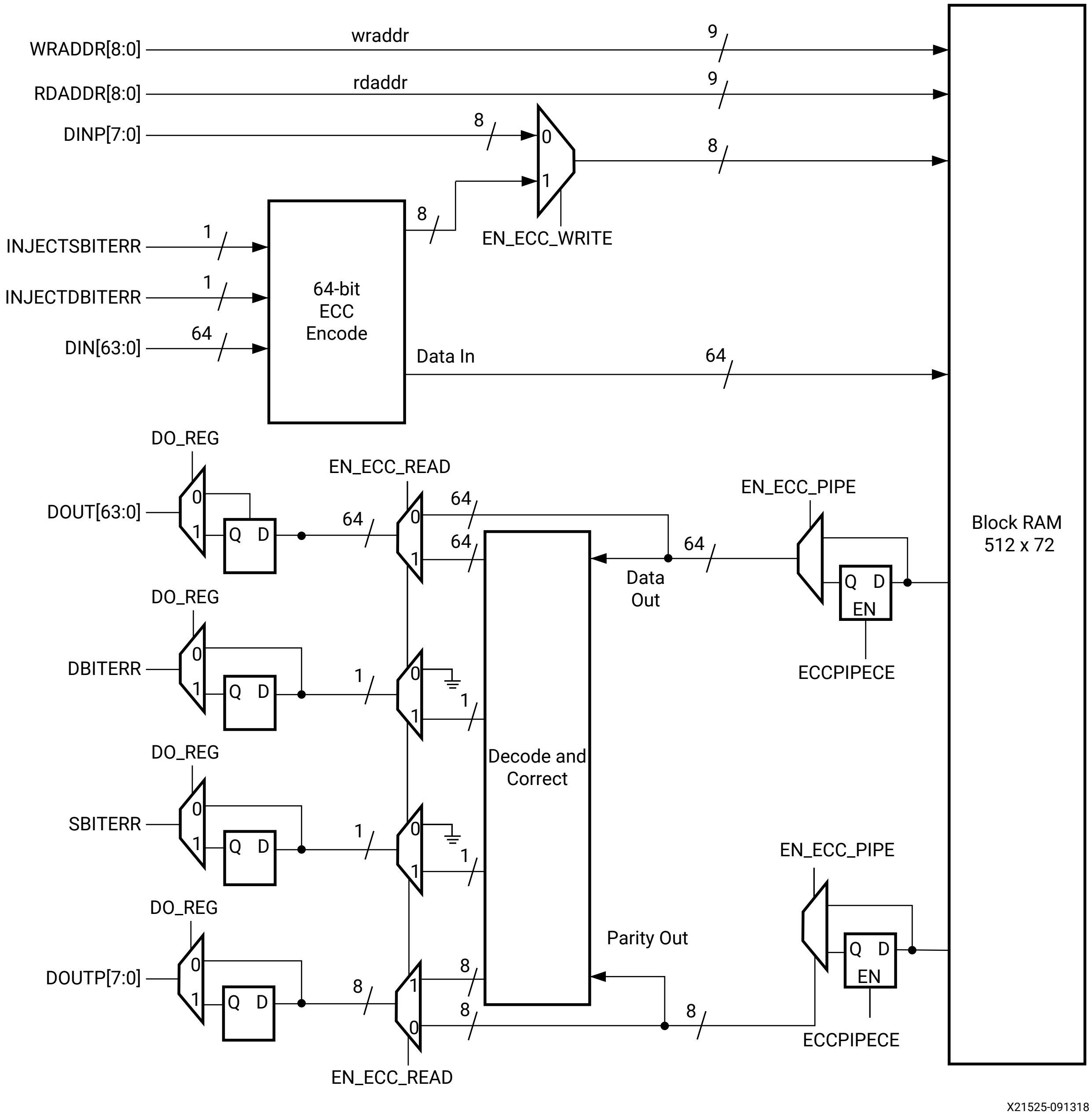
Block RAM ECC Attributes

Address Collision

Top-Level View of the Block RAM ECC Architecture

The following figure shows the top-level view of a block RAM in ECC mode.

Figure 17: **Top-Level View of Block RAM ECC**



Block RAM ECC Primitive

When using the RAMB36E5 in ECC mode, the input and output pins are identical to the block RAM primitive and tables described earlier in this document. In addition to the pin names shown there, the block RAM primitive has pins for use in ECC mode. The subsequent tables describe the pins used in ECC mode only.

Related Information

Block RAM Library Primitives

Block RAM Attributes

Block RAM ECC Port Descriptions

This table lists and describes the block RAM ECC-related I/O port names.

Table 20: **RAMB36E5 ECC Port Names and Descriptions**

Port Name	Signal Description
INJECTSBITERR	Injects a single-bit error if ECC is used. Creates a single-bit error at a particular block RAM bit location when asserted during write. The block RAM ECC logic corrects this error when this location is read back. The error is created in bit DIN[30].
INJECTDBITERR	Injects a double-bit error if ECC is used. Creates a double-bit error at two particular block RAM bit locations when asserted during write. The block RAM ECC logic flags a double-bit error when this location is read back. When both INJECTSBERR and INJECTDBERR signals are simultaneously asserted, a double-bit error is injected. The errors are created in bits DIN[30] and DIN[62].
SBITERR	ECC single-bit error output status. See also the dedicated cascade pins in this table when using the block RAM in ECC cascade mode.[1]
DBITERR	ECC double-bit error output status. See also the dedicated cascade pins in this table when using the block RAM in ECC cascade mode.[1]
CASINSBITERR	ECC single-bit error input in cascade mode. Cascade SBERR error bit status from the previous block RAM.
CASOUTSBITERR	ECC single-bit error output in cascade mode. Cascade SBERR error bit status to the next block RAM.
CASINDBITERR	ECC double-bit error input in cascade mode. Cascade DBERR error bit status from the previous block RAM.
CASOUTDBITERR	ECC double-bit error output in cascade mode. Cascade DBERR error bit status to the next block RAM.
ECCPIPECE	ECC pipeline register clock enable when EN_ECC_PIPE = TRUE. This is available only in ECC mode when EN_ECC_READ = TRUE.

Notes:

1. Hamming code implemented in the block RAM ECC logic detects one of three conditions: no detectable error, single-bit error detected and corrected on DOUT (but not corrected in the memory), and double-bit error detected without correction. SBITERR and DBITERR indicate these three conditions.

Block RAM ECC Attributes

This table lists the block RAM ECC attributes.

Table 21: **RAMB36E5 Attributes related to ECC**

Attribute Name	Type	Values	Default	Notes
EN_ECC_WRITE	Boolean	TRUE, FALSE	FALSE	Set to TRUE to enable ECC encoder.
EN_ECC_READ	Boolean	TRUE, FALSE	FALSE	Set to TRUE to enable ECC decoder.

Table 21: **RAMB36E5 Attributes related to ECC** *(cont'd)*

Attribute Name	Type	Values	Default	Notes
EN_ECC_PIPE	Boolean	TRUE, FALSE	FALSE	

ECC Modes of Operation

There are three types of ECC operation: standard, encode only, and decode only. The standard ECC mode uses both the encoder and decoder.

Standard ECC

Set by Attributes

EN_ECC_READ = TRUE

EN_ECC_WRITE = TRUE

ECC Encode Only

Set by Attributes

EN_ECC_READ = FALSE

EN_ECC_WRITE = TRUE

ECC Encode-Only Read

ECC encode-only read is identical to normal block RAM read. The 64-bit data appears at DOUT[63:0] and 8-bit parity appears at DOUTP[7:0]. Single-bit error correction does not occur, and the error flags SBITERR and DBITERR are never asserted.

ECC Decode Only

Set by Attributes

EN_ECC_READ = TRUE

EN_ECC_WRITE = FALSE

In ECC decode-only mode, only the ECC decoder is enabled. The ECC encoder is disabled. Decode-only mode is used to inject single-bit or double-bit errors to test the functionality of the ECC decoder. The ECC parity bits must be externally supplied using the DINP[7:0] pins.

Creating 8 Parity Bits for a 64-bit Word

Using logic external to the block RAM (a large number of XOR circuits), 8 parity bits can be created for a 64-bit word. However, using ECC encoder-only mode, the 8 parity bits can be automatically created without additional logic by writing any 64-bit word into a separate block RAM. The encoded 8-bit ECC parity data is immediately available, or the complete 72-bit word can be read out.

Block RAM ECC VHDL and Verilog Templates

VHDL and Verilog templates are available in the Vivado Design Suite.

Chapter 3

UltraRAM Resources

UltraRAM Key Features

The UltraRAM key features are:

- 288K bits of storage in a single block.
- Dual port, 32k x 9, 16k x 18, 8k x 36, and 4K x 72, single clock synchronous memory.
- UltraRAM cascade for building larger blocks. UltraRAM has dedicated routing resources for appropriate inputs and outputs to cascade from lower UltraRAM to upper UltraRAM.
- Error correction coding (ECC) on both ports with single bit error detection and correction and double bit error detection.
- Sleep power saving features.
- Automatic power savings through automatic invocation/release of the sleep mode in a chain of UltraRAM blocks.
- Optional pipeline flip-flops on the inputs, outputs, and cascade paths.
- Data out reset capability for outputs to be reset to all 0's.
- Single port widths of x9, x18, x36, and x72.
- Initialize to zeros or to user-defined values.

UltraRAM Cascade

The UltraRAM cascade features are:

- UltraRAM has dedicated routing resources for most of the inputs and outputs to cascade from lower UltraRAM to upper UltraRAM.
- Built-in address decode logic for 11 bits of the MSB address is used when cascading UltraRAMs to automatically generate internal enable for the read and write operations.
- Cascading in a column is supported in one direction from bottom to top and can be implemented without using general interconnect resources.

- Cascading between columns requires the use of device routing and potentially logic resources at the entry and exit points of each column.

UltraRAM Error Correction Coding

One 64-bit ECC block is provided per port per UltraRAM block. Independent encode and decode functionality is also available. The ECC mode can inject errors. The UltraRAM ECC features are:

- Optional ECC encode and decode on both ports.
- Single and double bit error detection.
- Single bit error correction.
- Single bit or double bit error injection capability.
- Optional pipeline register before and after ECC decode logic for maximum performance.

Block RAM and UltraRAM Differences

The key differences between block RAM and UltraRAM are:

- UltraRAM has one single clock input, is fully synchronous and, unlike the block RAM, does not support independent clock interfaces directly.
- UltraRAM can only support read or write per port per cycle.
- The simple dual-port (SDP) and true dual-port (TDP) block RAM modes do not directly apply to the UltraRAM. The UltraRAM port behavior can be viewed as a superset of SDP, but not TDP.
- Fixed read behavior; there are no user definable read-first, write-first, no-change modes with UltraRAM.
- Static data cascading; there are no dynamic cascade input or output multiplexer controls with UltraRAM.
- Address collision is not possible with UltraRAM.
- UltraRAM cascades data, address, and control signals, and not just the data lines.
- During the UltraRAM power saving mode (SLEEP), user operations are ignored and content corruption is not possible as long as setup and hold times are met. The memory content is preserved in the sleep power saving mode.

- Automatic power savings can be achieved by using the auto sleep feature that independently controls the wake-up and the sleep mode based on activity. This mode dynamically turns sleep mode on or off for selected UltraRAM blocks in a chain by predicting the activity of many cascaded UltraRAM blocks in a column or across several columns. For single UltraRAM block applications, using this feature would require many cycles of inactivity to be beneficial.

***Note*:** The setup/hold time of the UltraRAM address, enable, and sleep pins must not be violated. Violating the setup/hold time on these pins (even if write enable is Low) can corrupt the data contents of the UltraRAM.

Block RAM and UltraRAM Comparison

The following table shows a comparison of the block RAM and UltraRAM main features.

Table 22: **Block RAM and UltraRAM Comparison**

Feature	Block RAM	UltraRAM
Clocking	Two clocks	Single clock
Data width	Configurable (9, 18, 36, 72)	Configurable 9, 18, 36, 72
Modes	SDP and TDP	Two ports, each can independently read or write (a superset of SDP)
ECC	64-bit SECDED Supported in 64-bit SDP only (one ECC decoder for port A and one ECC encoder for port B)	64-bit SECDED One set of complete ECC logic for each port to enable independent ECC operations (ECC encoder and decoder for both ports)
Cascade	Cascade output only (input cascade implemented via logic resources) Cascade within a single clock region	Cascade both input and output (with global address decoding) Cascade across clock regions in a column Cascade across several columns with minimal logic resources
Power savings	One mode via manual signal assertion	One mode via manual signal assertion

UltraRAM Primitives

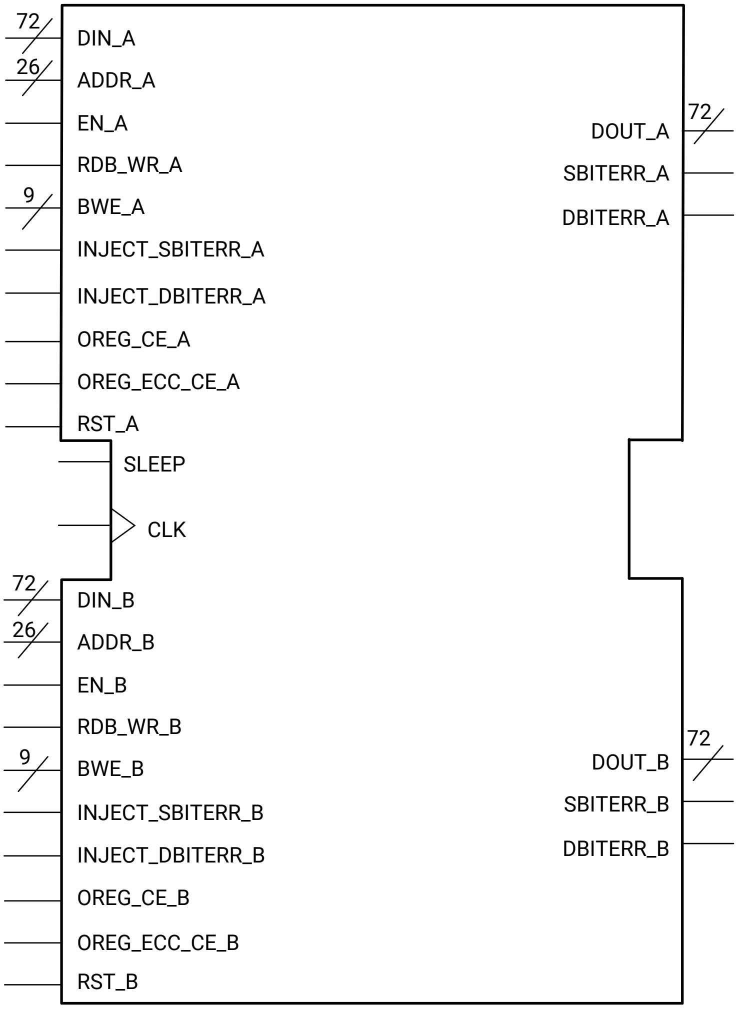
The following table lists the UltraRAM primitives.

Table 23: **UltraRAM Primitive**

Primitive	Description
URAM288E5	The URAM288E5 primitive supports all possible configurations including cascade and ECC.
URAM288E5_BASE	The URAM288E5_BASE primitive is a subset and supports single UltraRAM block instances without cascade capability.

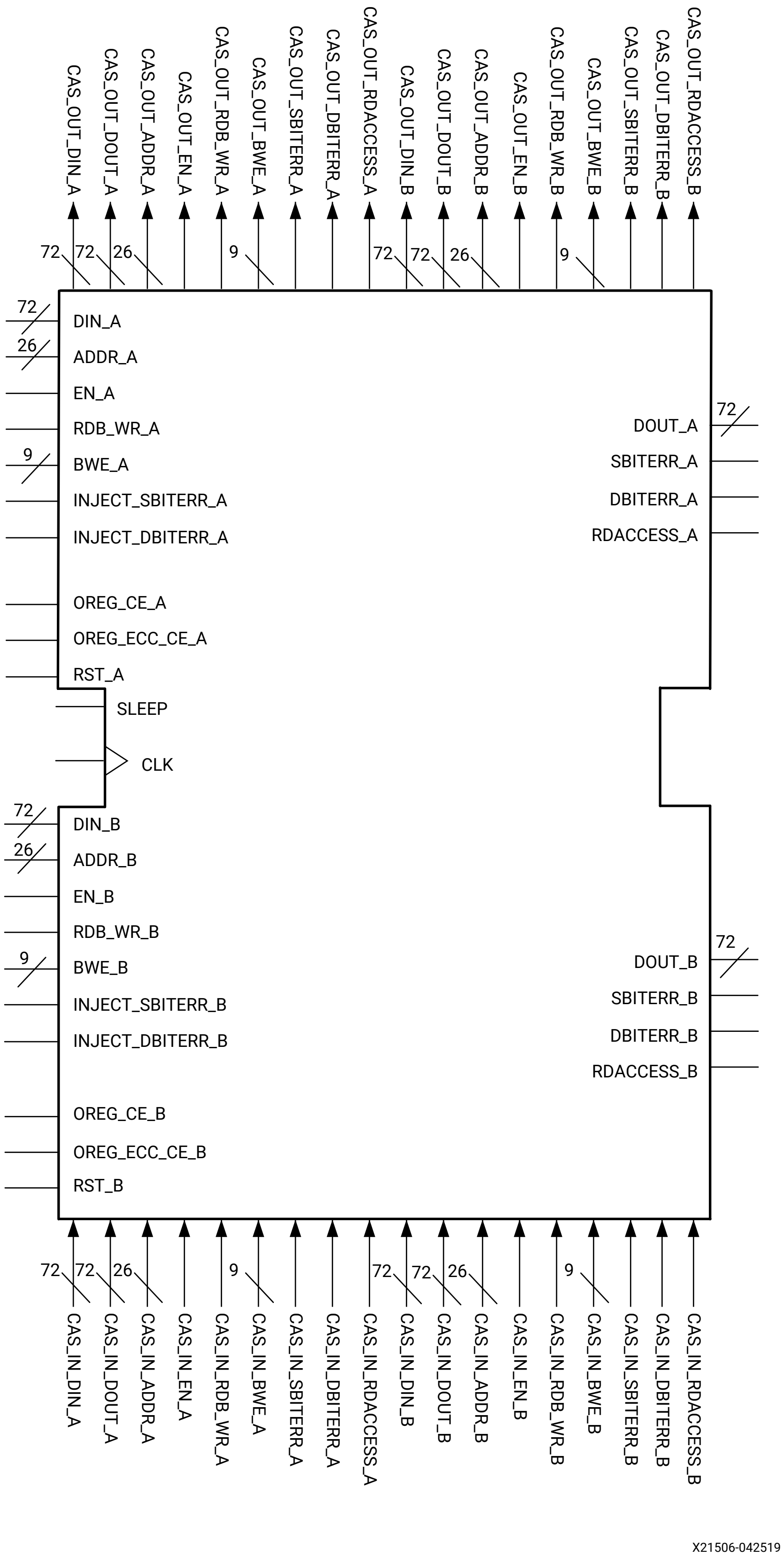
The UltraRAM URAM288E5 and URAM288E5_BASE library primitives are the basic building blocks for all UltraRAM configurations. The URAM288E5 primitive supports all possible configurations including cascade and ECC. The URAM288E5_BASE primitive is a subset and supports single UltraRAM block instances without cascade capability. The URAM288E5_BASE primitive is shown in Figure 18 and the URAM288E5 primitive is shown in Figure 19.

Figure 18: **UltraRAM URAM288E5_BASE Primitive**



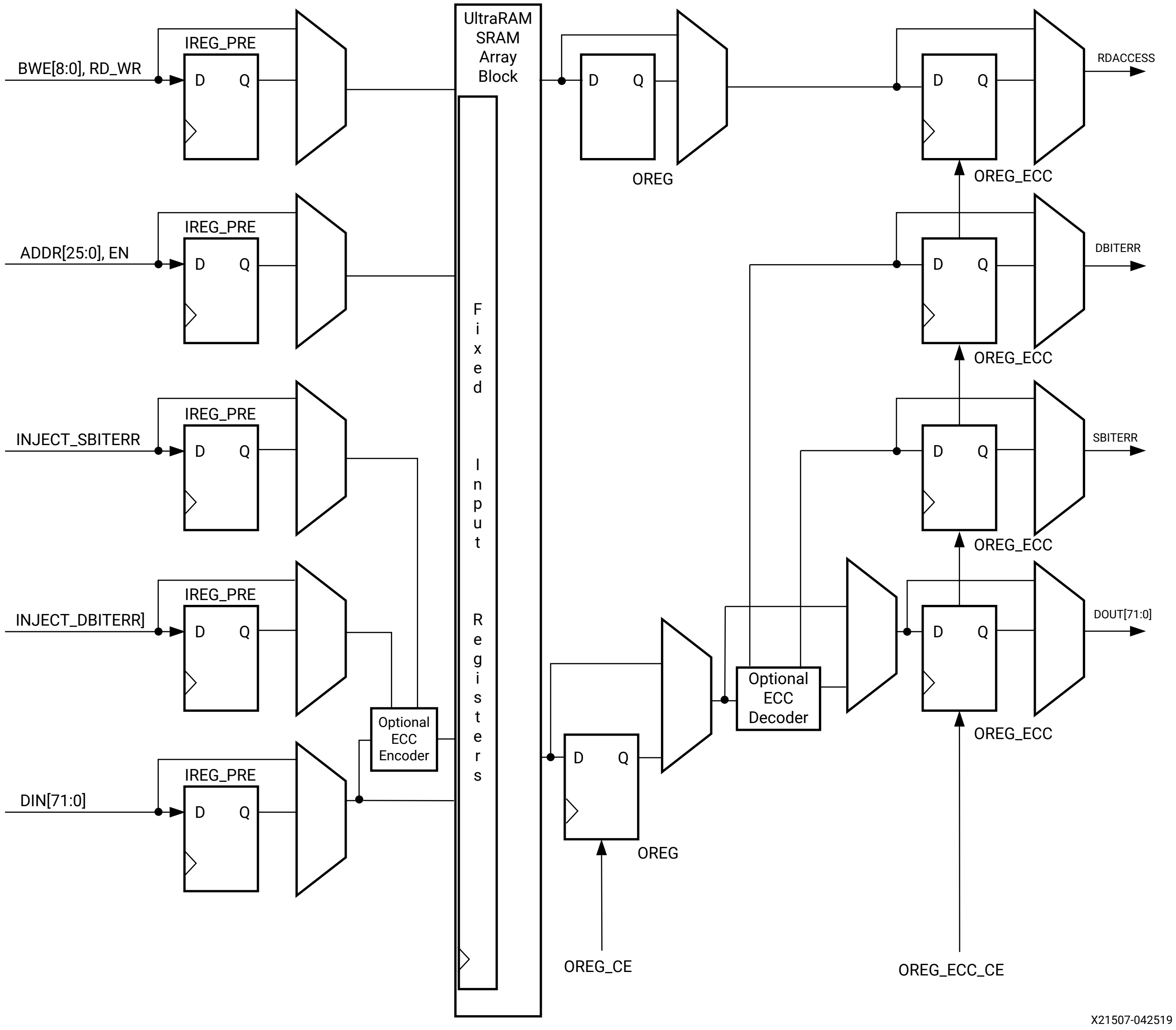
X21505-101518

Figure 19: **UltraRAM URAM288E5 Primitive**



The following figure depicts the simplified single UltraRAM block diagram without cascade with one port shown.

Figure 20: **Simplified Single UltraRAM Block Diagram without Cascade (One Port Shown)**



UltraRAM Port Names and Description

This section describes the UltraRAM port names.

No Cascade Ports

The following table lists the UltraRAM no cascade ports.

Table 24: **No Cascade Ports**

Port Name	Description
CLK	UltraRAM clock source.
SLEEP	Dynamic power gating control.
Port A Inputs	
ADDR_A[25:0]	Port A address. ADDR_A[25:15] are only used in cascade mode.
EN_A	Port A enable. Enables or disables the read/write access to the block RAM memory core.
RDB_WR_A	Port A read or write mode input select. Read (BAR) is active-Low (0 = read and 1 = write).
BWE_A[8:0]	Port A byte write enable.
DIN_A[71:0]	Port A write data in.
INJECT_SBITERR_A	Port A single-bit error injection during write.
INJECT_DBITERR_A	Port A double-bit error injection during write.
OREG_CE_A	Port A SRAM array core block read output pipeline register CLK enable.
OREG_ECC_CE_A	Port A ECC decoder output pipeline register CLK enable.
RST_A	Asynchronous or synchronous reset for port A output registers. Reset has priority over CE.
Port A Outputs	
DOUT_A[71:0]	Port A read data out.
RDACCESS_A	Port A read status output.
SBITERR_A	Port A single-bit error output status.
DBITERR_A	Port A double-bit error output status.
Port B Inputs	
ADDR_B[25:0]	Port B address. ADDR_B[25:15] are only used in cascade mode.
EN_B	Port B enable. Enables or disables the read/write access to the block RAM memory core.
BWE_B[8:0]	Port B byte write enable.
DIN_B[71:0]	Port B write data in.
INJECT_SBITERR_B	Port B single-bit error injection during write.
INJECT_DBITERR_B	Port B double-bit error injection during write.
OREG_CE_B	Port B SRAM array core block read output pipeline register CLK enable.
OREG_ECC_CE_B	Port B ECC decoder output pipeline register CLK enable.
RST_B	Asynchronous or synchronous reset for port B output registers. Reset has priority over CE.
Port B Outputs	
DOUT_B[71:0]	Port B read data out.
RDACCESS_B	Port B read status output.
SBITERR_B	Port B single-bit error output status.
DBITERR_B	Port B double-bit error output status.

Cascade Ports

The following table lists the UltraRAM cascade ports. Input ports are cascaded from the block below and output ports are cascaded to the block above.

Table 25: **Cascade Ports**

Port Name	Description
Port A Cascade Inputs	
CAS_IN_ADDR_A[25:0]	Port A input address input. In cascade mode, connect this port to CAS_OUT_ADDR_A.
CAS_IN_EN_A	Port A input enable input. In cascade mode connect this port to CAS_OUT_EN_A.
CAS_IN_BWE_A[8:0]	Port A input write mode port byte write enable. In cascade mode, connect this port to CAS_OUT_BWE_A.
CAS_IN_RDB_WR_A	Port A input read/write mode select. In cascade mode, connect this port to CAS_OUT_RDB_WR_A.
CAS_IN_DIN_A[71:0]	Port A input write mode. In cascade mode, connect this port to CAS_OUT_DIN_A.
CAS_IN_DOUT_A[71:0]	Port A input read mode data output. In cascade mode, connect this port to CAS_OUT_DOUT_A.
CAS_IN_RDACCESS_A	Port A input read mode read status. In cascade mode, connect this port to CAS_OUT_RDACCESS_A.
CAS_IN_SBITERR_A	Port A input read mode single-bit error flag input. In cascade mode, connect this port to CAS_OUT_SBITERR_A.
CAS_IN_DBITERR_A	Port A input read mode double-bit error flag input. In cascade mode, connect this port to CAS_OUT_SBITERR_A.
Port A Cascade Outputs	
CAS_OUT_ADDR_A[25:0]	Port A output address. In cascade mode, connect this port to CAS_IN_ADDR_A.
CAS_OUT_EN_A	Port A output enable. In cascade mode, connect this port to CAS_IN_EN_A.
CAS_OUT_RDB_WR_A	Port A output read/write mode select. In cascade mode, connect this port to CAS_IN_RDB_WR_A.
CAS_OUT_BWE_A[8:0]	Port A output write mode byte write enable. In cascade mode, connect this port to CAS_IN_BWE_A.
CAS_OUT_DIN_A[71:0]	Port A output write mode data. In cascade mode, connect this port to CAS_IN_DIN_A.
CAS_OUT_DOUT_A[71:0]	Port A output read mode data. In cascade mode, connect this port to CAS_IN_DOUT_A.
CAS_OUT_RDACCESS_A	Port A output read mode read status flag. In cascade mode, connect this port to CAS_IN_RDACCESS_A.
CAS_OUT_SBITERR_A	Port A output read single-bit error flag. In cascade mode, connect this port to CAS_IN_SBITERR_A.
CAS_OUT_DBITERR_A	Port A output read mode double-bit error flag. In cascade mode, connect this port to CAS_IN_DBITERR_A.
Port B Cascade Inputs	
CAS_IN_ADDR_B[25:0]	Port B input address input. In cascade mode, connect this port to CAS_OUT_ADDR_B.
CAS_IN_EN_B	Port B input enable input. In cascade mode, connect this port to CAS_OUT_EN_B.
CAS_IN_BWE_B[8:0]	Port B input write mode port byte write enable. In cascade mode, connect this port to CAS_OUT_BWE_B.
CAS_IN_RDB_WR_B	Port B input read/write mode select. In cascade mode, connect this port to CAS_OUT_RDB_WR_B.
CAS_IN_DIN_B[71:0]	Port B input write mode. In cascade mode, connect this port to CAS_OUT_DIN_B.

Table 25: **Cascade Ports** *(cont'd)*

Port Name	Description
CAS_IN_DOUT_B[71:0]	Port B input read mode data output. In cascade mode, connect this port to CAS_OUT_DOUT_B.
CAS_IN_RDACCESS_B	Port B input read mode read status. In cascade mode, connect this port to CAS_OUT_RDACCESS_B.
CAS_IN_SBITERR_B	Port B input read mode single-bit error flag input. In cascade mode, connect this port to CAS_OUT_SBITERR_B.
CAS_IN_DBITERR_B	Port B input read mode double-bit error flag input. In cascade mode, connect this port to CAS_OUT_DBITERR_B.
Port B Cascade Outputs	
CAS_OUT_ADDR_B[25:0]	Port B output address. In cascade mode, connect this port to CAS_IN_ADDR_B.
CAS_OUT_EN_B	Port A output enable. In cascade mode, connect this port to CAS_IN_EN_B.
CAS_OUT_BWE_B[8:0]	Port B output write mode byte write enable. In cascade mode, connect this port to CAS_IN_BWE_B.
CAS_OUT_RDB_WR_B	Port B output read/write mode select. In cascade mode, connect this port to CAS_IN_RDB_WR_B.
CAS_OUT_DIN_B[71:0]	Port B output write mode data. In cascade mode, connect this port to CAS_IN_DIN_B.
CAS_OUT_DOUT_B[71:0]	Port B output read mode data. In cascade mode, connect this port to CAS_IN_DOUT_B.
CAS_OUT_RDACCESS_B	Port B output read mode read status flag. In cascade mode, connect this port to CAS_IN_RDACCESS_B.
CAS_OUT_SBITERR_B	Port B output read single-bit error flag. In cascade mode, connect this port to CAS_IN_SBITERR_B.
CAS_OUT_DBITERR_B	Port B output read mode double-bit error flag. In cascade mode, connect this port to CAS_IN_DBITERR_B.

UltraRAM Port Signals

The following describes the UltraRAM port signals.

Clock – CLK

Each port is fully synchronous with a single clock pin for both ports. All port input pins have the setup time referenced to this CLK pin. The output data bus has a clock-to-out time referenced to the CLK pin. Clock polarity is configurable (rising edge by default).

Power Gating Enable Input – SLEEP

The dynamic power gating capability can be used to save static power when the memory is not actively used for extended periods of time.

When sleep mode is asserted, and setup and hold times are met, the memory starts going into sleep mode in the next clock cycle. The SLEEP inputs disable the UltraRAM read and write operation. Consequently, if a read or write operation is attempted, it is ignored until after the wake-up time is satisfied. However, setup and hold times must be met. While in sleep mode, the output of the SRAM array and the OREG pipeline registers are synchronously reset to 0 with the next rising edge of clock. The other optional pipeline registers are not affected by the sleep mode. Therefore, the ultimate data output value of the UltraRAM is either held at its previous value or appears to be reset to 0 depending on the usage of the other pipeline registers. The output of the OREG register is held to 0 until the first valid read data (after wake-up time) flows through the pipeline.

The SLEEP pin controls the power gating of the RAM. When SLEEP = 1, the SRAM peripheral logic is powered down to save energy. The data in the SRAM array is retained but it cannot be read from or written to. SLEEP allows a two clock cycles wake-up time with no impact on SEU performance. The polarity of this pin is not configurable (active-High).

Wake-up time defines when the EN pin can be asserted after SLEEP has been deasserted. The clock wake-up cycles mentioned previously assume no optional pipelines are enabled.

***Note*:** If the OREG is used (OREG=TRUE) and a read operation is followed immediately by a SLEEP operation (SLEEP going active), then the read operation data does not exit the UltraRAM block because the OREG pipeline stage is powered down immediately. The RDACCESS signal is still asserted although the data is not observed at the output.

Address Bus – ADDR_A, ADDR_B

The 26-bit address bus selects the memory cells for read or write. The lower 15 bits, The 3 LSB bits for port width selection, and the next 12 bits are used to select memory cells within the 4K location in each UltraRAM. The upper 11 bits select the UltraRAM that is used for cascading multiple UltraRAMs to form deep memory arrays. Each UltraRAM has a built-in comparator, which compares the upper 11 bit address with a unique SELF_ADDR attribute to identify if the UltraRAM has been selected. The SELF_MASK attribute defines how many of the 11 bit addresses should be used for the compare.

Enable – EN_A and EN_B

The enable pin affects the read and write functionality of the port. Ports with an inactive enable pin keep the output pins in the previous state and do not write data to the memory cells. Enable polarity is invertible (active High by default). However, during reset or the power saving mode (SLEEP), the outputs are reset to "0".

Read/Write Select – RDB_WR_A and RDB_WR_B

When this pin is 1, it selects the write operation. When it is 0, it selects the read operation. The polarity is invertible. Each port can only execute a read or write operation in one cycle. When a write operation is executed, the read outputs hold the previous value.

Byte-Wide Write Enable – BWE_A, BWE_B

Byte-wide write enable (BWE) is a 9-bit bus. Depending on the BWE_MODE_A/B attribute setting, bit 9 (BWE[8]) might not be used. In PARITY_INTERLEAVED mode, only the eight least significant bits are used. A single parity bit for each of the DIN bytes 0–7 corresponds to and is written to the related parity bit in the MSB of the DIN bus. This mode supports a custom parity scheme. In PARITY_INDEPENDENT mode, the nine BWE bits correspond to a byte of the DIN bus. Table 26 details how the BWE_A/B bits can be used to enable the corresponding DIN bits during a write operation. The byte write enable inputs are ignored during a read operation.

Note: All byte write enable bits must be set to 1 in ECC mode for proper operation.

Data-In Buses – DIN_A, DIN_B

Data-in buses provide the new data value to be written into UltraRAM. The data bus is 72-bits wide with the lower 64 bits used for data and the upper 8 bits used for parity or for regular data inputs.

Table 26: **Byte Write Enable (URAM288E5)**

BWE_MODE_A/B	BWE_A/B	DIN_A/B
Parity Interleaved Mode		
PARITY_INTERLEAVED	BWE_A/B[7]	DIN_A/B[71,63:56]
PARITY_INTERLEAVED	BWE_A/B[6]	DIN_A/B[70,55:48]
PARITY_INTERLEAVED	BWE_A/B[5]	DIN_A/B[69,47:40]
PARITY_INTERLEAVED	BWE_A/B[4]	DIN_A/B[68,39:32]
PARITY_INTERLEAVED	BWE_A/B[3]	DIN_A/B[67,31:24]
PARITY_INTERLEAVED	BWE_A/B[2]	DIN_A/B[66,23:16]
PARITY_INTERLEAVED	BWE_A/B[1]	DIN_A/B[65,15:8]
PARITY_INTERLEAVED	BWE_A/B[0]	DIN_A/B[64,7:0]
Parity Independent Mode		
PARITY_INDEPENDENT	BWE_A/B[8]	DIN_A/B[71:64]
PARITY_INDEPENDENT	BWE_A/B[7]	DIN_A/B[63:56]
PARITY_INDEPENDENT	BWE_A/B[6]	DIN_A/B[55:48]
PARITY_INDEPENDENT	BWE_A/B[5]	DIN_A/B[47:40]
PARITY_INDEPENDENT	BWE_A/B[4]	DIN_A/B[39:32]
PARITY_INDEPENDENT	BWE_A/B[3]	DIN_A/B[31:24]
PARITY_INDEPENDENT	BWE_A/B[2]	DIN_A/B[23:16]
PARITY_INDEPENDENT	BWE_A/B[1]	DIN_A/B[15:8]
PARITY_INDEPENDENT	BWE_A/B[0]	DIN_A/B[7:0]

Inject Single and Double Bit Error Inputs – INJECT_SBITERR_A, INJECT_DBITERR_A, INJECT_SBITERR_B, INJECT_DBITERR_B

The inject error inputs can induce a single or double bit error on write data input for testing purposes.

Register Enable for OREG Pipeline Stage – OREG_CE_A, OREG_CE_B

This register enable pin controls the first optional output register. When this register is enabled using the OREG_A/B attribute, and the corresponding CE input is High, the read data is stored in the register at the rising clock edge. The polarity of CE inputs is not configurable (active-High).

Register Enable for OREG_ECC Pipeline Stage – OREG_ECC_CE_A, OREG_ECC_CE_B

This register enable pin controls the ECC optional output register. When this register is enabled using the OREG_ECC_A/B attribute, and the corresponding CE input is High, the read data is stored in the register at the rising clock edge. The polarity of CE inputs is not configurable (active-High).

Reset – RST_A, RST_B

There are two modes for the reset operation. The synchronous and asynchronous reset modes are controlled by the RST_MODE_A/B attributes. In synchronous reset mode, which is the default, all output flip-flops and latches are synchronously reset to 0. In the asynchronous reset mode, all output flip-flops and latches are reset to 0 without waiting for a CLK edge. This operation does not affect UltraRAM memory cells and does not disturb write operations on either of the ports. The polarity for both signals is configurable (active-High by default).

When used in an UltraRAM matrix, the RST input is expected to be asserted (and deasserted) simultaneously at the input of all UltraRAMs in the matrix (in both SYNC and ASYNC reset modes). Consequently, after a RST operation, a new read data is available after N cycles (where N is the read latency of the matrix). However, if the read operation overlaps with the reset operation, the DOUT could change from the reset value to a new read value earlier than the N cycles. This occurs because the read output corresponding to read during or before reset might propagate to the output (because input pipelines IREG_PRE/IREG_CAS are not impacted by the reset). Consequently, this behavior also depends on the REG_CAS locations in the matrix. When the REG_CAS location changes, the DOUT behavior after reset can be different. See the figures in Read/Write Waveforms With Reset – With and Without Optional Output Pipeline Registers for the timing diagrams showing an example of this difference in behavior.

***Note*:** When in asynchronous reset mode, the UltraRAM does not have any built-in synchronizers on this input for the deassertion edge. Cascaded UltraRAM use cases need a common synchronizer (typically implemented at an upper level of hierarchy). The address and control input signal from the fabric must be properly synchronized before it is supplied to the UltraRAM, otherwise memory corruption can occur due to a violation of a setup or hold time.

Data-Out Buses – DOUT_A, DOUT_B

Data-out buses reflect the contents of memory cells referenced by the address bus at the last active clock edge during a read operation. During a write operation or no operation, data-out buses are not changed and the data is preserved from the previous cycle. This applies to both single UltraRAM and cascade/matrix configurations. Similarly for a cascaded UltraRAM, the read output at the end of the cascade chain (at exit point) also holds the previous data. The data bus is 72-bits wide with the lower 64 bits used for data and the upper 8 bits used for parity or as regular data outputs.

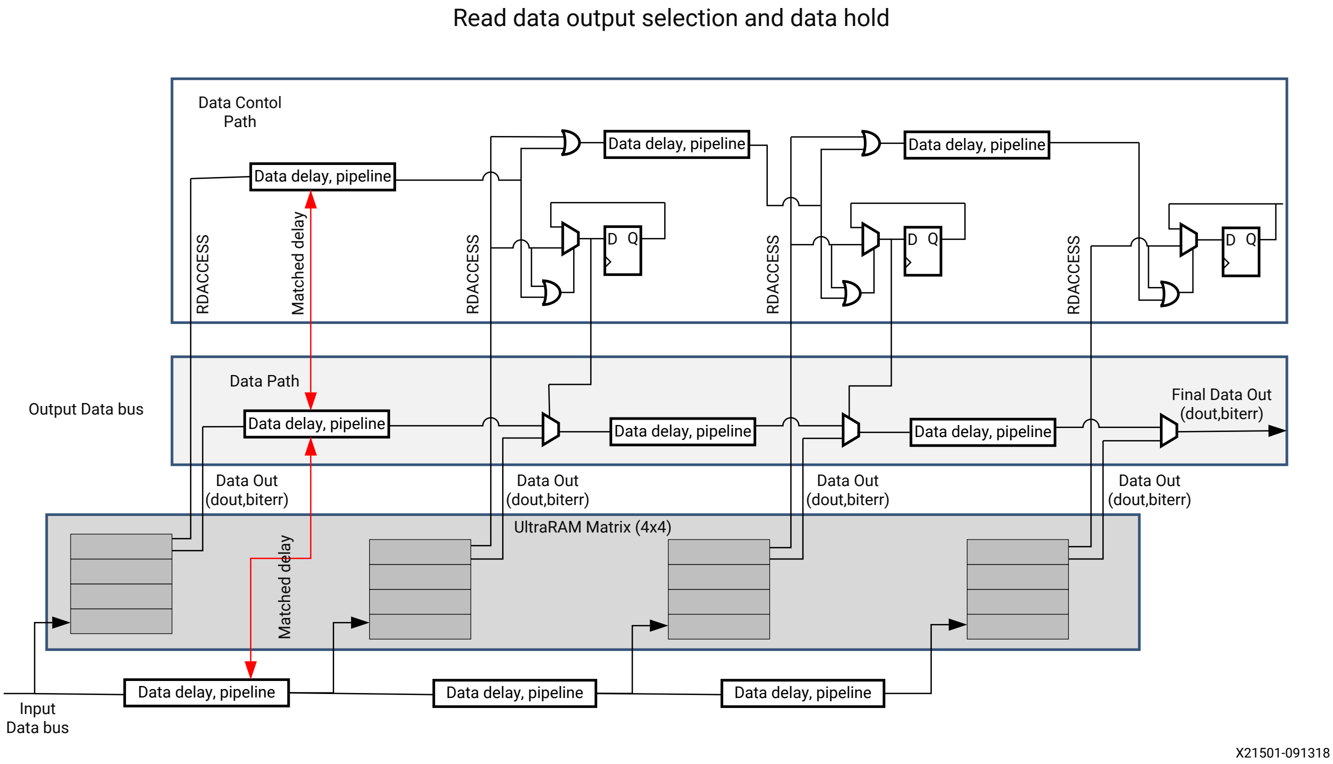
Read Status Output – RDACCESS_A, RDACCESS_B

The UltraRAM generates a read access status output (RDACCESS_A/B) to indicate that a read operation finished executing, indicating when new data is available at the output. This output has the same latency as the corresponding read data. This output can then be used at the top level to select the correct read data when cascading UltraRAMs across multiple columns. When this output is High, it indicates a read operation has been executed in that UltraRAM or in an UltraRAM below it that is part of the cascade chain. When crossing columns of cascaded UltraRAMs, CLB registers might be required to account for pipelining in the column cascade.

The main purpose of the RDACCESS signal is to support UltraRAMs that are arranged in a matrix fashion. It identifies which UltraRAM block in a matrix configuration is actively reading data in a given clock cycle. The application can then determine the appropriate read data that needs to be propagated to the final output for processing.

The following figure illustrates a use case where the RDACCESS signal is used to select the correct read output data and control the output data path of a matrix. The circuit holds the data for the inactive outputs. It is important to match the input delay between the matrix entry point (for example, horizontal pipelining in the fabric for performance reasons) with identical delays on the output side (the delay/pipeline blocks shown in the figure).

Figure 21: **RDACCESS Signal Use Case**



ECC Error Bit Output – SBITERR_A, DBITERR_A, SBITERR_B, DBITERR_B

The ECC error bit outputs are valid when EN_ECC_RD_A/B attributes are set to TRUE. These outputs are asserted when the ECC decoder identifies a single bit error or a double bit error.

Invertible Control Signal Pins

The five control pins CLK, EN_A/B, RST_A/B each have an individual inversion option. EN and RST control signals can be configured as active-High or active-Low, and the clock can be active on a rising or falling edge (active-High on a rising edge is the default) without requiring other logic resources.

UltraRAM Attributes

This table describes the UltraRAM attributes.

Table 27: **UltraRAM Attributes**

Attributes	Values	Default	Type	Description
AUTO_SLEEP_LATENCY	3 to 15	8	DECIMAL	Sets the latency requirement for UltraRAM to sleep mode.
AVG_CONS_INACTIVE_CYCLES	10 to 100000	10	DECIMAL	Sets the average consecutive inactive cycles in sleep mode. When in sleep mode, this is defined as the average number of cycles with no read/write operation on either port. Used by the power reporting tools. Set by the user.
BWE_MODE_A	PARITY_INTERLEAVED, PARITY_INDEPENDENT	PARITY_INTERLEAVED	STRING	Port A byte write control for either 1 byte/1 bit 8 bytes/1 byte parity. For WRITE_WIDTH less than 72 bits, the PARITY_INDEPENDENT mode is not supported.
BWE_MODE_B	PARITY_INTERLEAVED, PARITY_INDEPENDENT	PARITY_INTERLEAVED	STRING	Port B byte write control for either 1 byte/1 bit 8 bytes/1 byte parity. For WRITE_WITDH less than 72 bits, the PARITY_INDEPENDENT mode is not supported.
CASCADE_ORDER_CTRL_A	NONE, FIRST, MIDDLE, LAST	NONE	STRING	Port A position of UltraRAM block in the cascade chain. Controls ADDR, EN and RDB_WR.
CASCADE_ORDER_CTRL_B	NONE, FIRST, MIDDLE, LAST	NONE	STRING	Port B position of UltraRAM block in the cascade chain. Controls ADDR, EN and RDB_WR.
CASCADE_ORDER_DATA_A	NONE, FIRST, MIDDLE, LAST	NONE	STRING	Port A position of UltraRAM block in the cascade chain for data. Controls DIN, BWE, DOUT, RDACCESS, and SBITERR/DBITERR. When CASCADE_ORDER_DATA = FIRST or MIDDLE, the DOUT, SBITERR, DBITERR, and RDACCESS outputs should not be used.
CASCADE_ORDER_DATA_B	NONE, FIRST, MIDDLE, LAST	NONE	STRING	Port B position of UltraRAM block in the cascade chain for data. Controls DIN, BWE, DOUT, RDACCESS, and SBITERR/DBITERR. When CASCADE_ORDER_DATA = FIRST or MIDDLE, the DOUT, SBITERR, DBITERR, and RDACCESS outputs should not be used.

Table 27: **UltraRAM Attributes** *(cont'd)*

Attributes	Values	Default	Type	Description
EN_AUTO_SLEEP_MODE	FALSE, TRUE	FALSE	STRING	Enables UltraRAM to automatically go into power saving mode.
EN_ECC_RD_A	FALSE, TRUE	FALSE	STRING	Port A ECC decoder used for data read or not.
EN_ECC_RD_B	FALSE, TRUE	FALSE	STRING	Port B ECC decoder used for data read or not.
EN_ECC_WR_A	FALSE, TRUE	FALSE	STRING	Port A ECC encoder used for data write or not.
EN_ECC_WR_B	FALSE, TRUE	FALSE	STRING	Port B ECC encoder used for data write or not.
INIT_000 to INIT_3FF	Any 288-bit Hex value	`288'h000...`	HEX	Initializes the content of the memory array during configuration.
INIT_FILE	Any string	"NONE"	STRING	Memory initialization file.
IREG_PRE_A	FALSE, TRUE	FALSE	STRING	Inserts port A data, address, and control input pipeline registers.
IREG_PRE_B	FALSE, TRUE	FALSE	STRING	Inserts port B data, address, and control input pipeline registers.
IS_CLK_INVERTED	FALSE, TRUE	FALSE	STRING	Optional inverter for CLK.
IS_EN_A_INVERTED	FALSE, TRUE	FALSE	STRING	Port A optional inverter for EN.
IS_EN_B_INVERTED	FALSE, TRUE	FALSE	STRING	Port B optional inverter for EN.
IS_RDB_WR_A_INVERTED	FALSE, TRUE	FALSE	STRING	Port A optional inverter for RDB_WR.
IS_RDB_WR_B_INVERTED	FALSE, TRUE	FALSE	STRING	Port B optional inverter for RDB_WR.
IS_RST_A_INVERTED	FALSE, TRUE	FALSE	STRING	Port A optional inverter for reset input.
IS_RST_B_INVERTED	FALSE, TRUE	FALSE	STRING	Port B optional inverter for reset input.
MATRIX_ID	Custom label	NONE	STRING	Custom label (string) to set a matrix ID name used by the power reporting tools to tag all of the UltraRAM blocks that belong to a cascade chain or matrix. Assign different names to each matrix. Single UltraRAM instances do not require a label. Used by the power reporting tools. Set by the user or synthesis tools.

Table 27: **UltraRAM Attributes** *(cont'd)*

Attributes	Values	Default	Type	Description
NUM_URAM_IN_MATRIX	1 to 2048	1	DECIMAL	Defines the cascade/ matrix size (the number of UltraRAMs in a matrix). Attach to the instances in a particular matrix. For single instances, set to 1. Used by the power reporting tools. Set by the user or synthesis tools.
NUM_UNIQUE_SELF_ADDR_A	1 to 2048	1	DECIMAL	The number of unique SELF_ADDR_A UltraRAM blocks in a cascade chain or matrix. Typically equal to the number of blocks in a cascade chain or matrix. In the broadcast case, the number could be smaller due to common SELF_ADDR_A settings. Used by the power reporting tools. Set by the user or synthesis tools.
NUM_UNIQUE_SELF_ADDR_B	1 to 2048	1	DECIMAL	The number of unique SELF_ADDR_B UltraRAM blocks in a cascade chain or matrix. Typically equal to the number of blocks in a cascade chain or matrix. In the broadcast case, the number is smaller due to common SELF_ADDR_B settings. Used by the power reporting tools. Set by the user or synthesis tools.
OREG_A	FALSE, TRUE	FALSE	STRING	Inserts port A SRAM array output optional pipeline register.
OREG_B	FALSE, TRUE	FALSE	STRING	Inserts port B SRAM array output optional pipeline register.
OREG_ECC_A	FALSE, TRUE	FALSE	STRING	Inserts port A ECC decoder output optional pipeline register.
OREG_ECC_B	FALSE, TRUE	FALSE	STRING	Inserts port B ECC decoder output optional pipeline register.
PR_SAVE_DATA	FALSE, TRUE	FALSE	STRING	Enables skipping of content initialization after PR to maintain previous memory content.

Table 27: **UltraRAM Attributes** *(cont'd)*

Attributes	Values	Default	Type	Description
READ_WIDTH_A	9, 18, 36, 72	72	DECIMAL	Specifies the desired data width for a read on Port A, including parity bits. This attribute should be set to the same value for every UltraRAM in a cascade chain.
READ_WIDTH_B	9, 18, 36, 72	72	DECIMAL	Specifies the desired data width for a read on Port B, including parity bits. This attribute should be set to the same value for every UltraRAM in a cascade chain.
REG_CAS_A	FALSE, TRUE	FALSE	STRING	Inserts port A cascade data input and data output pipeline registers.
REG_CAS_B	FALSE, TRUE	FALSE	STRING	Inserts port B cascade data input and data output pipeline registers.
RST_MODE_A	SYNC, ASYNC	SYNC	STRING	Port A reset mode.
RST_MODE_B	SYNC, ASYNC	SYNC	STRING	Port B reset mode.
SELF_ADDR_A	`11'h000 to 11'h7ff`	`11'h000`	HEX	Port A self-address value.
SELF_ADDR_B	`11'h000 to 11'h7ff`	`11'h000`	HEX	Port B self-address value.
SELF_MASK_A	`11'h000 to 11'h7ff`	`11'h7ff`	HEX	Port A self-address mask.
SELF_MASK_B	`11'h000 to 11'h7ff`	`11'h7ff`	HEX	Port B self-address mask.
USE_EXT_CE_A	FALSE, TRUE	FALSE	STRING	Port A attribute to allow either internal or external control for the CE pins on all output pipeline registers.
USE_EXT_CE_B	FALSE, TRUE	FALSE	STRING	Port B attribute to allow either internal or external control for the CE pins on all output pipeline registers.
WRITE_WIDTH_A	9, 18, 36, 72	72	DECIMAL	Specifies the desired data width for a write to Port A, including parity bits. This attribute should be set to the same value for every UltraRAM in a cascade chain.
WRITE_WIDTH_B	9, 18, 36, 72	72	DECIMAL	Specifies the desired data width for a write to Port B, including parity bits. This attribute should be set to the same value for every UltraRAM in a cascade chain.

***Note*:** The URAM288E5_BASE primitive does not have any of the cascade attributes.

Auto Sleep Latency – AUTO_SLEEP_LATENCY

The auto sleep mode automatically uses the function provided by the SLEEP pin in an automated manner. To determine when to go to sleep and when to wake up, the UltraRAM looks ahead in terms of RAM access.

When the EN_AUTO_SLEEP_MODE attribute is set to TRUE, the AUTO_SLEEP_LATENCY attribute defines the number of clock cycles the enable and global address inputs EN_A, EN_B, ADDR_A[25:15], and ADDR_B[25:15] have to arrive prior to other inputs. This lookahead information is used to decide when the UltraRAM can go to sleep. The EN_A/B and ADDR_A/B[25:15] are internally delayed to implement this feature. The number of clock cycles required for the enable and global address signals to arrive early is set with the AUTO_SLEEP_LATENCY attribute, which can take values between 3 – 15. Therefore, for the UltraRAM to go into sleep mode, a minimum number of consecutive inactive clock cycles is required that is defined by the value of the AUTO_SLEEP_LATENCY attribute.

The number of sleep cycles achieved is calculated with this formula:

- If the number of consecutive inactive cycles is < AUTO_SLEEP_LATENCY, then the number of sleep cycles = 0.
- If the number of consecutive inactive cycles is ≥ AUTO_SLEEP_LATENCY, then the number of consecutive sleep cycles = the number of consecutive inactive cycles – 3.
- An inactive cycle is defined as a cycle where there is no RD/WR operation from either port.

For example, to obtain sleep cycles in any five or more consecutive cycles of inactivity, set the AUTO_SLEEP_LATENCY to five.

Once in auto sleep mode, the output of the OREG retains the old value for only one clock cycle. The data can be reset to 0 in the very next cycle or later depending on when the UltraRAM enters sleep mode. If there are no other pipeline registers enabled after OREG, the design must use the register output read data only during the last valid clock cycle. If there are other consecutive pipeline registers, then those pipeline registers will hold the last read data.

The auto sleep mode is most effective for large chain sizes or any chain with very little activity. While there is a default for this attribute, the application determines the effective power savings of this feature based on activity, latency, and other needs of the application. If AUTO_SLEEP_LATENCY is too low, the UltraRAM goes into sleep and wake-up too often, which can cause more power to be consumed. If it is too high, then the maximum of amount of power might not be saved.

Byte Write Enable Mode – BWE_MODE_[A|B]

This attribute determines the data and parity usage of the byte write enable (BWE_[A/B]) inputs. Either the one data byte\one parity bit mode is selected (PARITY_INTERLEAVED) or the eight data bytes/1 parity byte mode is selected (PARITY_INDEPENDENT).

Related Information

Byte-Wide Write Enable – BWE_A, BWE_B

Cascade Chain Order – CASCADE_ORDER_CTRL[A|B], CASCADE_ORDER_DATA_[A/B]

These attributes indicate if an UltraRAM is part of a cascade and the location of the UltraRAM in the cascade chain with respect to control and data flow. The values are NONE (default), FIRST, MIDDLE, and LAST. All UltraRAMs that are first in each column should be set to FIRST, and all UltraRAMs that are last in each column should be set to LAST. All UltraRAMs in between must be set to MIDDLE.

Related Information

Cascade User Attributes

Cascading UltraRAM and Matrix Configurations

AVG_CONS_INACTIVE_CYCLES, MATRIX_ID, NUM_URAM_IN_MATRIX, and NUM_UNIQUE_SELF_ADDR_A|B Attributes

These attributes have no functional impact on the design and are used for power estimation and power reporting. The Vivado tools set these attributes automatically when the UltraRAMs are inferred through synthesis.

***Note*:** When manually instantiated, the attributes should be set to reflect the actual usage for accurate power calculations. If not set, the power estimates will be pessimistic.

***Note*:** In the broadcast use case, the SELF_MASK setting can affect the NUM_UNIQUE_SELF attribute. While the ADDR_A/B[25:15] inputs can be different on UltraRAM instances in a matrix, the SELF_MASK setting might result in a match for multiple instances in a matrix. See example shown in Figure 26.

Set Enable Auto Sleep Mode – EN_AUTO_SLEEP_MODE

This mode enables auto sleep mode for automatic power savings and overwrites the application's control of sleep mode. When set to TRUE, the user-controlled sleep input is disabled. Instead, UltraRAM internally puts itself in and out of sleep mode to achieve automatic power savings. The criteria for UltraRAM to go in and out of sleep mode is a measure of inactive clock cycles determined by the AUTO_SLEEP_LATENCY attribute.

***Note*:** The use of USE_EXT_CE_A/B is not permitted when in AUTO_SLEEP mode. Both attributes are not permitted to be true at the same time.

Enable ECC Write – EN_ECC_WR_[A/B]

This attribute determines if the ECC encoder (write) is enabled or not.

Enable ECC Read – EN_ECC_RD_[A/B]

This attribute determines if the ECC decoder (read) is enabled or not.

INIT_XX - UltraRAM Content Initialization

The memory content can be initialized or cleared in the configuration bitstream with the INIT_0000 - INIT_3FFF (1024) attributes or by reading a file with the INIT_FILE attribute. Each INIT_XXX is a 72-digit hexadecimal-encoded bit vector. The memory contents can be partially initialized and are automatically completed with zeros. Each INIT_XXX attribute contains 32 bits of parity from bit locations 287:256 and 256 bits of data from bit locations 255:0 as shown in the following tables. Used UltraRAM blocks not initialized in the design are automatically initialized with 0 during configuration and are not part of the bitstream. Therefore, the bitstream size varies depending on the number of user-initialized UltraRAM blocks.

Table 28: **UltraRAM INIT Attribute (General Format)**

Attribute	Parity Location	Data Location
	X72 - D[71:64] * 4	X72 - D[63:0] * 4
	X36 - D[35:32] * 8	X72 - D[31:0] * 8
	X18 - D[17:16] * 16	X72 - D[15:0] * 16
	X9 - D[8] * 32	X72 - D[7:0] * 32
INIT_001	287:256	255:000
INIT_000	575:544	543:288
....		
INIT_3FF	294911:294889	294879:294624

Table 29: **UltraRAM Detailed Initialization**

INIT_000	x72	x36	x18	x9
Parity [287:256] A[15:3] = 12'h003 A[15:3] = 12'h002 A[15:3] = 12'h001 A[15:3] = 12'h000	D[71:64]	A[2] = 1, D[35:32]	A[2:1] = 11, D[17:16]	A[2:0] = 111, D[8]
				A[2:0] = 110, D[8]
			A[2:1] = 10, D[17:16]	A[2:0] = 101, D[8]
				A[2:0] = 100, D[8]
		A[2] = 0, D[35:32]	A[2:1] = 01, D[17:16]	A[2:0] = 011, D[8]
				A[2:0] = 010, D[8]
			A[2:1] = 00, D[17:16]	A[2:0] = 001, D[8]
				A[2:0] = 000, D[8]

Table 29: **UltraRAM Detailed Initialization** *(cont'd)*

INIT_000	x72	x36	x18	x9
Data [255:000] A[15:3] = 12'h003 A[15:3] = 12'h002 A[15:3] = 12'h001 A[15:3] = 12'h000	D[63:0]	A[2] = 1, D[31:0]	A[2:1] = 11, D[15:0]	A[2:0] = 111, D[7:0]
				A[2:0] = 110, D[7:0]
			A[2:1] = 10, D[15:0]	A[2:0] = 101, D[7:0]
				A[2:0] = 100, D[7:0]
		A[2] = 0, D[31:0]	A[2:1] = 01, D[15:0]	A[2:0] = 011, D[7:0]
				A[2:0] = 010, D[7:0]
			A[2:1] = 00, D[15:0]	A[2:0] = 001, D[7:0]
				A[2:0] = 000, D[7:0]

Optional Input Register Stage – IREG_PRE_[A|B]

This attribute determines if EN/RDB_WR/BWE/ADDR/DIN/INJECT_SBITERR/INJECT_DBITERR UltraRAM inputs have their respective input pipeline registers enabled or not. IREG_PRE and REG_CAS are mutually exclusive except as noted in Optional Cascade Register Stage – REG_CAS_[A|B]. See Figure 22.

Optional Output Register Stage – OREG_[A|B]

This attribute determines if the SRAM array output has a pipeline stage enabled or not.

Optional ECC Output Register Stage – OREG_ECC_[A|B]

This attribute determines if the ECC error and data outputs of the ECC decode logic have the pipeline registers enabled or not.

Optional Cascade Register Stage – REG_CAS_[A|B]

Determines if both cascade data/controls/address inputs and outputs have their pipeline registers (IREG_CAS and OREG_CAS) enabled or not. These pipeline stages play a critical role in determining the maximum frequency of the UltraRAM. In cascade mode, these registers should be used in each block or every few blocks depending on the maximum frequency requirement. IREG_PRE and REG_CAS are mutually exclusive except when the CASCADE_ORDER attribute is set to MIDDLE or LAST, the IREG_PRE register can still be used in the cascade case for the error injection inputs INJECT_S/DBITERR. For all other inputs REG_CAS must be used. See Figure 22.

Reset Mode – RST_MODE_[A|B]

Determines if RST_[A/B] is a synchronous or asynchronous input to CLK.

Self Address – SELF_ADDR_[A|B]

This attribute determines the self-address of the UltraRAM and must be a unique value for each UltraRAM in the cascade chain. This determines the address of each UltraRAM in the cascade chain. It is an 11-bit value and can have any value from `11'h000` to `11'h7ff`.

Related Information
Cascade User Attributes

Self Mask Value – SELF_MASK_[A|B]

This attribute determines how many bits in the cascaded address (SELF_ADDR) are used for the comparison (address decoding) with the ADDR input to determine if the input address matches the UltraRAM in the cascade chain. The number of address bits is determined by the total address space in the cascaded UltraRAMs. The MSB bits corresponding to the unused address bits should be set to 1.

Related Information
Cascade User Attributes

External CE Usage – USE_EXT_CE_[A|B]

This attribute enables the use of external CE inputs to control all the output pipeline stages in non-cascade mode. By default, the design uses the internally generated CEs to control all the pipeline stages. This does not apply to the OREG_CAS registers enables. In cascade mode, the OREG_CAS register enables are automatically controlled by the UltraRAM. Using the RDACCESS output signal is not allowed when external CE mode is enabled.

***Note*:** In cascade mode, USE_EXT_CE cannot be used and should be set to false. Consequently, in cascade mode, the external CE inputs (OREG_CE and OREG_ECC_CE) cannot be used. This attribute is only supported when CASCADE_ORDER=NONE.

Additional URAM288E5 Primitive Design Considerations

The following is a checklist of UltraRAM design considerations.

Optional Output Registers

Optional output registers can be used at either or both A|B output ports of URAM288E5. The choice is made using the OREG_[A|B] attribute. The two independent clock enable pins are OREG_CE_[A|B].

Similarly, optional ECC decoder output registers can be used at either or both A|B output ports. The choice is made using the OREG_ECC_[A|B] attribute. The two independent clock enable pins are OREG_ECC_CE_[A|B].

Figure 20 shows the optional output registers. Using the output registers is strongly recommended to enable a higher clock rate. However, each output register adds a clock cycle latency of one.

Output Register Reset

When using the optional output registers at port [A|B], assertion of the RST_[A|B]) pin resets the output registers. RST_MODE_[A|B] can be set to SYNC or ASYNC to determine whether RST_[A|B] is a synchronous or asynchronous. The RST_[A|B] has priority over OREG_CE_[A|B] and OREG_ECC_CE_[A|B].

Optional Sleep Mode

The UltraRAM supports dynamic power gating (sleep) modes for power savings when there is no access activity. The EN_AUTO_SLEEP_MODE attribute enables automated sleep mode by predicting the activity of many cascaded UltraRAM blocks, or the SLEEP input port can be asserted to explicitly put an UltraRAM into a sleep mode. Although the OREG_[A|B] output registers are reset when in sleep mode, the UltraRAM retains its data. SLEEP allows a two clock-cycle wake-up time.

Minimum Clock Pulse and Address/Enable/Sleep Setup/Hold Time (Caution: Unstable Clock)

IMPORTANT! *The clock minimum pulse width and setup/hold time of the UltraRAM address, enable, and sleep pins must not be violated. Violating the clock minimum pulse width or these setup/hold times (even if write enable is Low) can corrupt the data contents of the UltraRAM. This most commonly occurs during an unstable clock (for example, when unplugging an external clock source) or when flip-flops driving block RAM control pins are asynchronously reset, such as a system-wide reset. To avoid this issue, ensure stable clocks and design with synchronous resets for both assertion and deassertion. When the clock is not stable, disable the clock buffer or logic driving the UltraRAM control pins, or deassert the UltraRAM EN input.*

Dual Port SRAM Array Operations

The dual-port 288 Kb UltraRAM consists of a 288 Kb storage area and two independent access ports, A and B. Both ports share a single clock input.

In each clock cycle, each port can perform either a read or a write operation independent of the other port. Any combination of read/write is allowed on any of the two ports. The read and write operations are always synchronous to the clock. The operation of port A is always executed first followed by the operation of port B within the same clock cycle. Consequently, data access collision is not possible when both ports access the same address locations. Each port has its own address, data in, data out, enable, and write enable.

- If both ports are executing read and write for the same address, the behavior is defined as (see the following table):
  - If port A is writing, port B is reading, then port B reads new data.
  - If port A is reading, port B is writing, then port A reads the old data.
  - If port A and B are writing, then port B write overwrites the port A write. At the end of the clock cycle, the memory stores port B write data.

Table 30: **UltraRAM Port Access**

UltraRAM Port Access	Port A	Port B	Data Output
1 Read/1 Write	Read	Write	Old data
1 Read/1 Write	Write	Read	New data
1 Read/1 Read and Write	Read	Read/Write	Old data
1 Read/1 Read and Write	Read/Write	Read	New data
2 Read/2 Write	Read or Write	Write or Read	Depends on port A/B read/ write combination

Read Operation

In default mode with no optional pipeline registers enabled, the read operation uses one clock edge. The read address is registered on the read port, and the stored data is loaded into the output latch after the SRAM access time. When using additional optional input/output registers, the read operation needs extra cycles depending on how many pipeline registers are used. The read data is held on the output until the next valid read operation or until a reset operation changes the output.

Write Operation

A write operation is a single clock-edge operation, unless the optional input register is enabled. The write address is registered on the write port, and the data input is stored in memory. The read output holds the previous value during a write operation, unless the reset input is asserted.

Optional Input Registers

The optional data, address, and control input registers (IREG_PRE registers) improve design performance by eliminating the routing delay from the CLB flip-flops for pipelined operation. Optional input registers (IREG_CAS registers) for cascading data, address and control are available. Either the data input or the cascade input registers can be used at any given time for an UltraRAM block depending on its configuration (input cascaded or not). Both the input and output cascade registers are enabled via the REG_CAS attribute simultaneously and cannot be turned on or off individually.

Optional Output Registers

The optional output registers improve design performance by eliminating the routing delay to the CLB flip-flops for pipelined operation. The first optional output register (OREG stage) is immediately after the SRAM array read operation. Additional optional output registers after the ECC decode logic (OREG_ECC stage) and cascade logic (OREG_CAS register) are available. By default, the design uses internally generated CE to control all the pipeline stages for power saving. However, an external CE port can be used by setting the USE_EXT_CE_A/B attribute. When the external CE is enabled, an independent clock enable input port is provided for these output registers. If the output data registers are disabled via their CE port, they hold their value independent of the input register operation.

RESET Operation

An UltraRAM RST operation simultaneously resets the read data/status/ECC error outputs and all corresponding optional output/cascade pipeline registers. The reset function can be synchronous (the default) or asynchronous depending on the RST_MODE attribute setting. The reset operation has priority over any read operation and any of the CE inputs. After deasserting RST, the reset value is valid until a new read data value flows through the pipeline.

When in asynchronous reset mode, the UltraRAM does not have any built-in synchronizers on this input for deassertion. Therefore, a logic-based synchronizer might be required for the RST input.

Note: If the design utilizes cascade, then a common synchronizer should be used for all the UltraRAM RSTs in the chain.

Byte Write Enable Function

The byte write enable feature allows a single byte of the input data to be written to the SRAM array. There are nine bits of write enable inputs for each port A and port B. There are two modes of operation that are selected by the BWE_MODE_[A/B] attribute. In PARITY_INTERLEAVED mode, each write enable bit enables eight data bits plus one parity bit. So each byte has a corresponding single parity bit. In the PARITY_INDEPENDENT mode, each write enable bit (BWE[7:0] enables the writing of eight data bits (one byte). The BWE bit number nine (BWE[8]) enable bit controls the one byte of eight parity bits. The byte write inputs are ignored during a read operation.

Note: If the ECC feature is used, all byte write enable bits must be set to "1" for proper operation of the ECC encoder/decoder.

Cascading UltraRAM and Matrix Configurations

One of the advanced, built-in features of UltraRAM is the capability to build deeper RAMs by directly cascading UltraRAM blocks in a single column through a dedicated direct interconnect. Ports for data in, data out, ECC error, address, enables, read/write select, and a write mask attribute facilitate cascading (see Figure 22).

Cascading is supported in only one direction and is always in a bottom-up fashion. UltraRAM blocks can be cascade unlimited in a single column within an SLR without limitation and have built-in connections. Cascade pipeline registers (IREG_CAS and OREG_CAS stages are enabled by the REG_CAS attribute) are available options in each UltraRAM. These registers can be enabled as needed depending on the maximum frequency and latency requirements of the design. Cascading from one clock region to the next clock region above can require additional pipeline registers on both the input and output side of the cascade chain to avoid potential setup time violations.

Cascading UltraRAMs across different columns can be achieved using logic and routing resources. The UltraRAM generates a read access status output RDACCESS_A/B to indicate that a read operation was executed. This output has the same latency as the corresponding read data and can be used to determine the correct read data when cascading using multiple columns.

If there is no read operation being performed, the read output at the end of the cascade chain (at the block exit point) will hold the previous data.

UltraRAMs in Versal devices have two cascade options that can be selected by the CASCADE_ORDER attributes to configure the UltraRAMs into cascade chains for control and data flow. There are two CASCADE_ORDER attributes for each port. CASCADE_ORDER_CTRL determines the address, enable, and read/write operations of the blocks in the cascade chain. CASCADE_ORDER_DATA determines the flow of data through the cascade chain. The

CASCADE_ORDER_CTRL attribute controls ADDR, EN, and RDB_WR only while the CASCADE_ORDER_DATA attribute controls DIN, BWE, DOUT, RDACCESS, SBITERR, and DBITERR. In most applications the control and data cascades are of identical length. The data cascade can never extend beyond the control cascade chain length in order to avoid read collision. Refer to the following table for all legal combinations of the attributes.

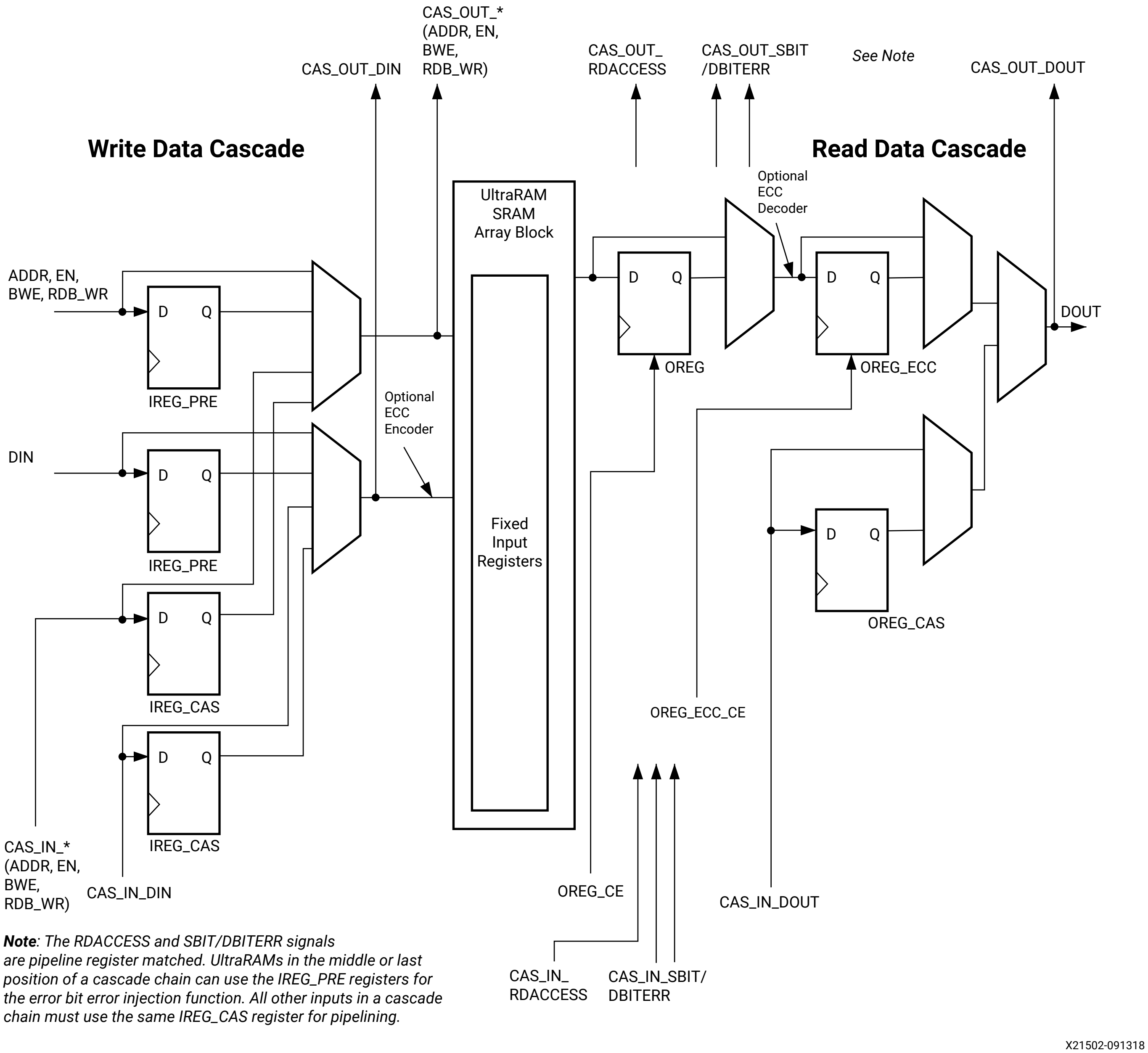
Table 31: **Legal Combinations of the CASCADE_ORDER Attributes**

CASCADE_ORDER_CTRL_A/B	CASCADE_ORDER_DATA_A/B
NONE	NONE
FIRST	NONE, FIRST
MIDDLE	NONE, FIRST, MIDDLE, LAST
LAST	NONE, LAST

Note that the IREG_PRE attribute is used to enable or disable input pipeline registers for inputs that have their corresponding CASCADE_ORDER_CTRL/DATA set to NONE or FIRST. Similarly, the REG_CAS attribute is used to enable or disable cascade pipeline registers when CASCADE_ORDER_CTRL/DATA is MIDDLE or LAST. When CASCADE_ORDER_CTRL and CASCADE_ORDER_DATA are set to different values, both IREG_PRE and REG_CAS attributes can be used to enable pipelines as required. In addition, IREG_PRE can also be used for bit error injection pins that are independent of the CASCADE_ORDER attributes.

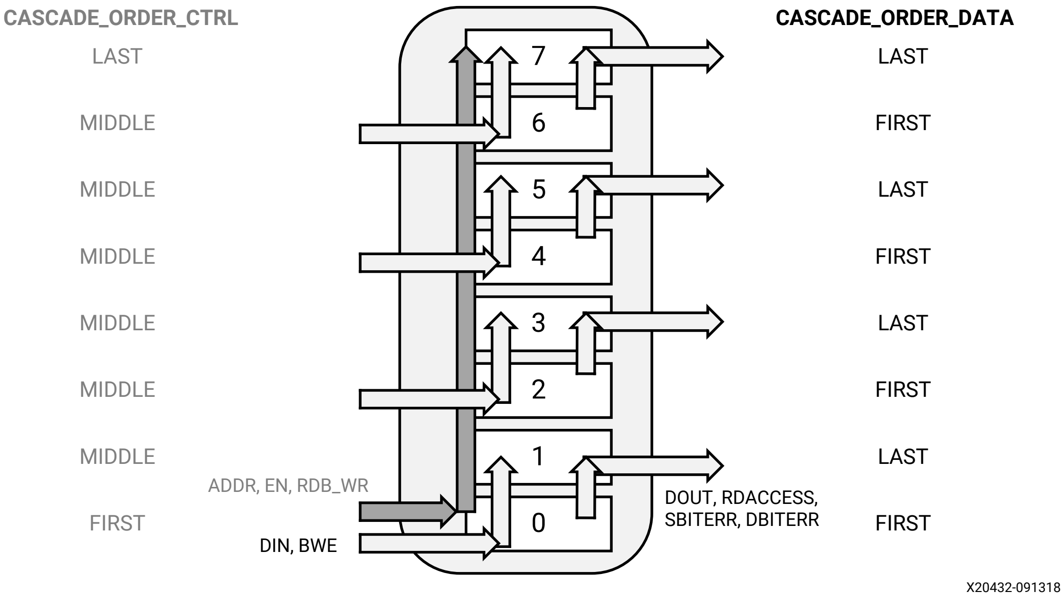
The OREG and OREG_ECC pipeline registers must be used identically in a data CASCADE_ORDER. In other words, the above pipeline register must all be either enabled or disabled in a CASCADE_ORDER_DATA chain between the FIRST and LAST UltraRAM block in the data chain.

Figure 22: **UltraRAM Cascade Block Diagram (One Port shown)**



***Note**: The RDACCESS and SBIT/DBITERR signals are pipeline register matched. UltraRAMs in the middle or last position of a cascade chain can use the IREG_PRE registers for the error bit error injection function. All other inputs in a cascade chain must use the same IREG_CAS register for pipelining.*

Figure 23: **UltraRAM Control and Data Cascade Block Diagram**



Cascade User Attributes

- CASCADE_ORDER_A/B determines the UltraRAM block cascade order for control and data.
  - NONE (Default) – UltraRAM is not in cascade mode.
  - FIRST – UltraRAM is the first instance in a cascade chain in each column of the chain.
  - MIDDLE – UltraRAM is a the middle instance in a cascade chain.
  - LAST – UltraRAM is the last instance in a cascade chain in each column of the chain.

  For legal combinations of cascade order attributes, see Table 31.

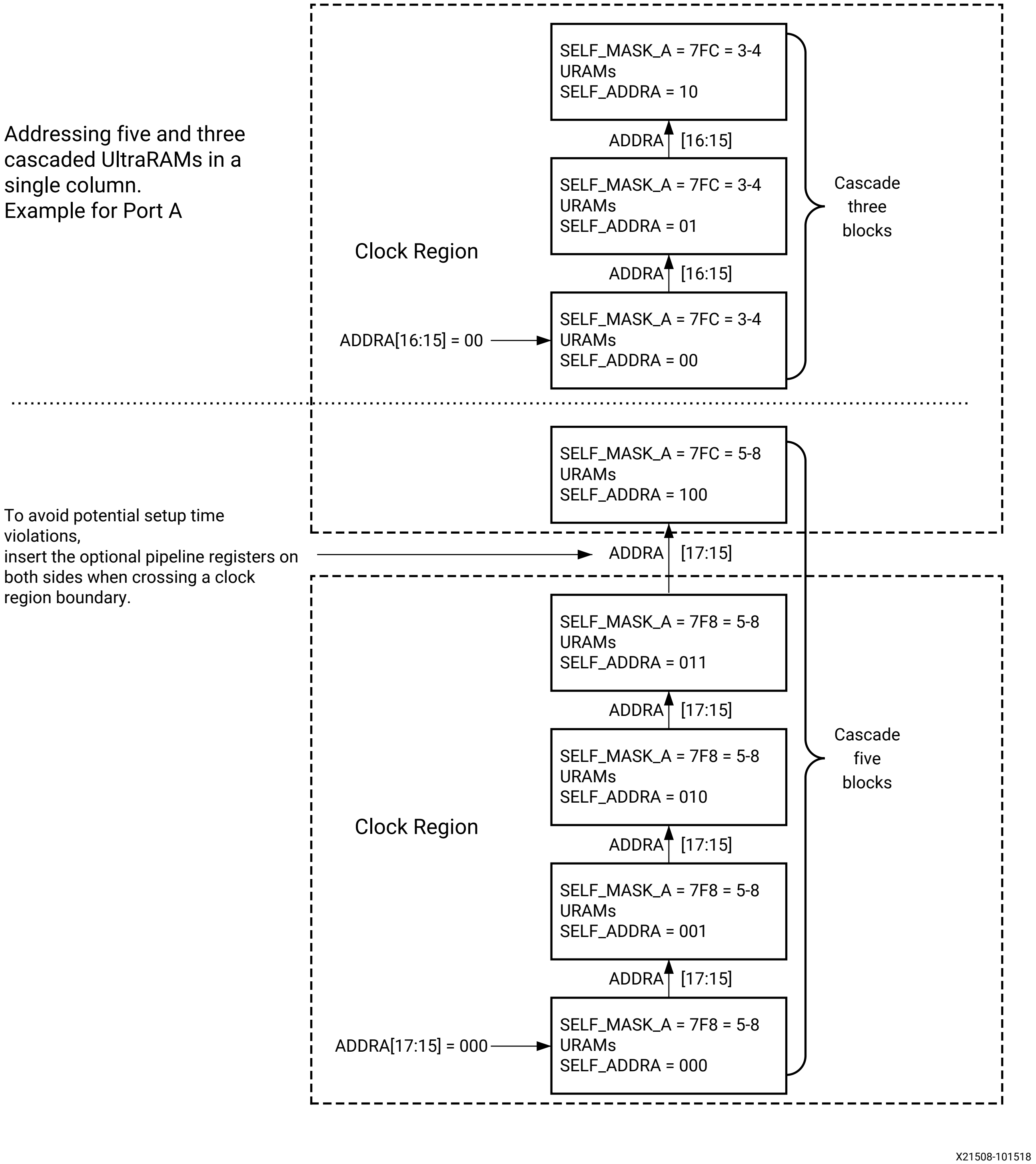
- SELF_MASK_A/B[25:15] determines the number of UltraRAM blocks in the cascade chain and therefore which of the ADDR_A/B[22:12] bits are used.
  - `11'h7ff` (Default) – Not in cascade mode. ADDR_A/B[25:15] inputs are masked.
  - `11'h7fe` – 2 UltraRAMs are cascaded. ADDR_A/B[25:16] inputs are masked.
  - `11'h7fc` – 3-4 UltraRAMs are cascaded. ADDR_A/B[25:17] inputs are masked
  - `11'h7f8` – 5-8 UltraRAMs are cascaded. ADDR_A/B[25:18] inputs are masked.
  - `11'h7f0` – 9-16 UltraRAMs are cascaded. ADDR_A/B[25:19] inputs are masked.
  - `11'h7e0` – 17-32 UltraRAMs are cascaded. ADDR_A/B[25:20] inputs are masked.
  - `11'h7c0` – 31-64 UltraRAMs are cascaded. ADDR_A/B[25:21] inputs are masked.

  - `11'h780` – 65-128 UltraRAMs are cascaded. ADDR_A/B[25:22] inputs are masked.
  - `11'h700` – 129-256 UltraRAMs are cascaded. ADDR_A/B[25:23] inputs are masked.
  - `11'h600` – 257-512 UltraRAMs are cascaded. ADDR_A/B[25:24] inputs are masked.
  - `11'h400` – 513-1024 UltraRAMs are cascaded. ADDR_A/B[25] input is masked.
  - `11'h000` – 1025-2048 (1036) UltraRAMs are cascaded. None of the address inputs are masked.

- SELF_ADDR_A/B[25:15]

  This attribute is used in cascade mode and must be set depending on which address bit in the UltraRAM cascade chain addresses the particular block to which it is attached. A particular UltraRAM block in the cascade chain is accessed when the self-address bit is set after masking with the address bits with the SELF_MASK_A/B attribute that matches the used ADDR_A/B address bits. The default is `11'h0` (see the following figure).

Figure 24: **Cascade Chain Examples**

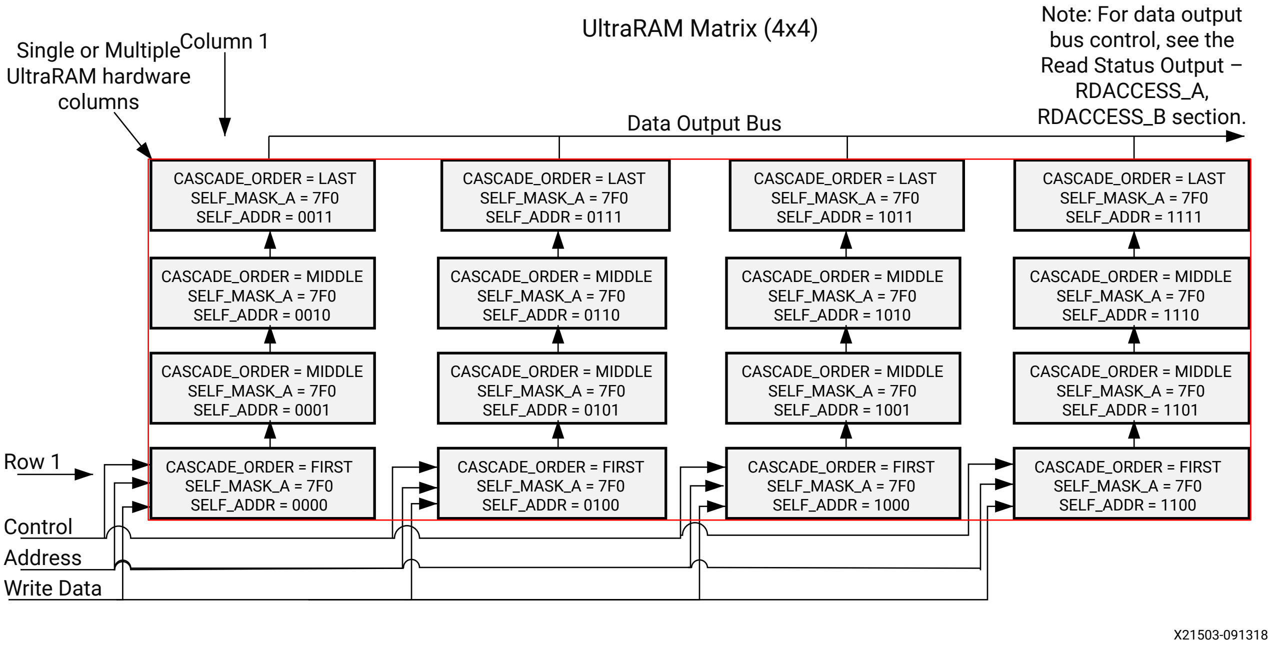

Addressing five and three cascaded UltraRAMs in a single column.
Example for Port A
Clock Region
SELF_MASK_A = 7FC = 3-4 URAMs
SELF_ADDRA = 10
ADDRA [16:15]
SELF_MASK_A = 7FC = 3-4 URAMs
SELF_ADDRA = 01
ADDRA [16:15]
ADDRA[16:15] = 00
SELF_MASK_A = 7FC = 3-4 URAMs
SELF_ADDRA = 00
Cascade three blocks
SELF_MASK_A = 7FC = 5-8 URAMs
SELF_ADDRA = 100
To avoid potential setup time violations,
insert the optional pipeline registers on both sides when crossing a clock region boundary.
ADDRA [17:15]
SELF_MASK_A = 7F8 = 5-8 URAMs
SELF_ADDRA = 011
ADDRA [17:15]
SELF_MASK_A = 7F8 = 5-8 URAMs
SELF_ADDRA = 010
Clock Region
ADDRA [17:15]
SELF_MASK_A = 7F8 = 5-8 URAMs
SELF_ADDRA = 001
ADDRA [17:15]
ADDRA[17:15] = 000
SELF_MASK_A = 7F8 = 5-8 URAMs
SELF_ADDRA = 000
Cascade five blocks
X21508-101518

Note: SELF_ADDR/SELF_MASK can be used even when CASCADE_ORDER_CTRL is set to NONE. This allows non-contiguous UltraRAMs to be cascaded using fabric logic if needed. Typically SELF_MASK | SELF_ADDR is unique for all the UltraRAMs in a cascade chain. This is a requirement for a read port to ensure the correct read data propagated to the exit point.

Building a Matrix From Cascaded UltraRAMs

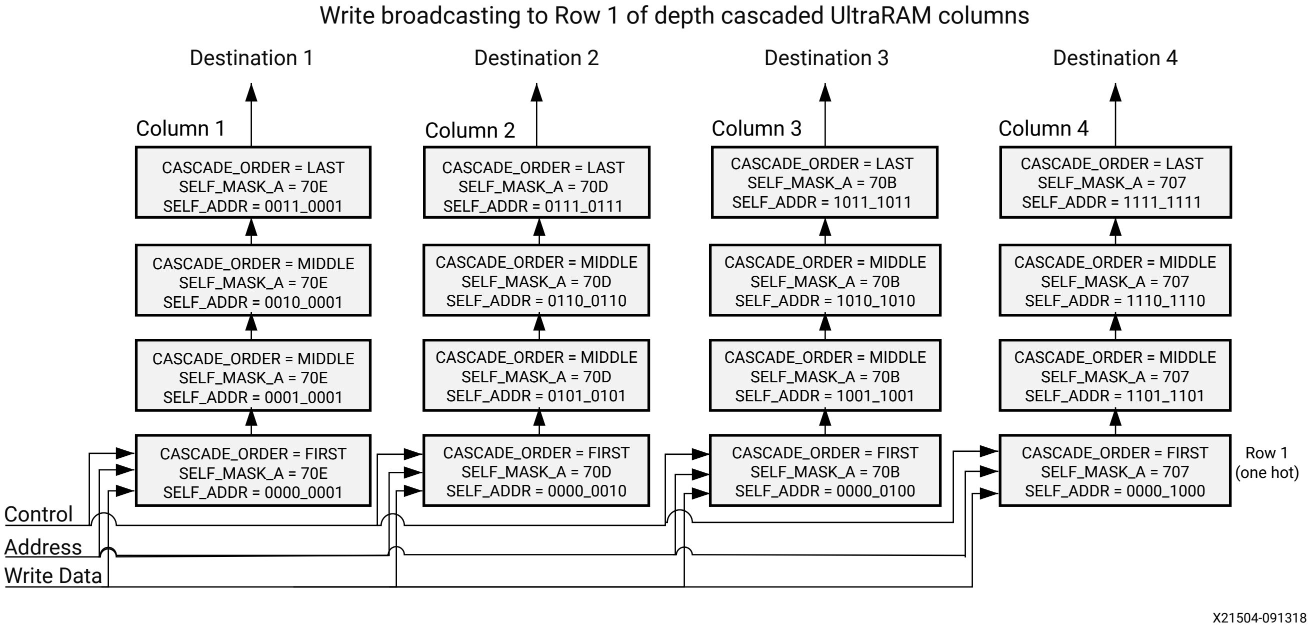
Applications can build a deep logical SRAM with multiple UltraRAMs. These UltraRAM instances form a matrix such that address, control signals, and input data arrive at the UltraRAM matrix at the bottom left and output data appear at the top right. The following figure illustrates the concept behind the UltraRAM matrix. In this X by Y (row x column) matrix, each matrix element is a single UltraRAM block cascaded vertically. To read/write from/to a matrix, address, control signals and input data (if write) enter the UltraRAM matrix in row 1. A write operation writes the input data to the addressed UltraRAM block at location row, column (R,C) and the word in it. Similarly, for a read operation, the output data reaches the output bus on top of the columns (always) by selecting an UltraRAM R,C and location in it. The figure illustrates a 4x4 UltraRAM matrix.

Figure 25: **4x4 UltraRAM Matrix**


Single or Multiple UltraRAM hardware columns
Column 1
UltraRAM Matrix (4x4)
Note: For data output bus control, see the Read Status Output – RDACCESS_A, RDACCESS_B section.
Data Output Bus
CASCADE_ORDER = LAST SELF_MASK_A = 7F0 SELF_ADDR = 0011
CASCADE_ORDER = LAST SELF_MASK_A = 7F0 SELF_ADDR = 0111
CASCADE_ORDER = LAST SELF_MASK_A = 7F0 SELF_ADDR = 1011
CASCADE_ORDER = LAST SELF_MASK_A = 7F0 SELF_ADDR = 1111
CASCADE_ORDER = MIDDLE SELF_MASK_A = 7F0 SELF_ADDR = 0010
CASCADE_ORDER = MIDDLE SELF_MASK_A = 7F0 SELF_ADDR = 0110
CASCADE_ORDER = MIDDLE SELF_MASK_A = 7F0 SELF_ADDR = 1010
CASCADE_ORDER = MIDDLE SELF_MASK_A = 7F0 SELF_ADDR = 1110
CASCADE_ORDER = MIDDLE SELF_MASK_A = 7F0 SELF_ADDR = 0001
CASCADE_ORDER = MIDDLE SELF_MASK_A = 7F0 SELF_ADDR = 0101
CASCADE_ORDER = MIDDLE SELF_MASK_A = 7F0 SELF_ADDR = 1001
CASCADE_ORDER = MIDDLE SELF_MASK_A = 7F0 SELF_ADDR = 1101
Row 1
CASCADE_ORDER = FIRST SELF_MASK_A = 7F0 SELF_ADDR = 0000
CASCADE_ORDER = FIRST SELF_MASK_A = 7F0 SELF_ADDR = 0100
CASCADE_ORDER = FIRST SELF_MASK_A = 7F0 SELF_ADDR = 1000
CASCADE_ORDER = FIRST SELF_MASK_A = 7F0 SELF_ADDR = 1100
Control
Address
Write Data
X21503-091318

With the individual address encoding scheme for each UltraRAM, each block individually determines if it should have data. SELF_ADDRESS and SELF_MASK allow for non-unique addresses in a matrix, which allows for the broadcasting of data to multiple UltraRAMs simultaneously in the same cycle. The SELF_ADDRESS can be used as a one-hot encoded address (even partially), and the SELF_MASK determines which address bits are important and which address bits can be ignored (one-cold). Consequently, a global address applied to all UltraRAMs in a matrix can apply to a set of predetermined UltraRAMs. The following figure illustrates a multicast write of data to two, more, or all UltraRAMs in row 1. In this example, the lower four block address bits are ORed via the SELF_ADDR settings for the block that must simultaneously receive data, while the SELF_MASK ignores address bits not to be decoded for a block. In this use case, the UltraRAM can only be used in 1 read/1 write mode.

Figure 26: **Write Broadcasting to Row 1 of Depth Cascaded UltraRAM Columns**



Address Bit Decoding

- Row 1 Multicast

  ADDR18:15 = `0011` multicast to columns 1 + 2

  ADDR18:15 = `0101` multicast to columns 1 + 3

  ADDR18:15 = `0110` multicast to columns 2 + 3

  ADDR18:15 = `0111` multicast to columns 1 + 2 + 3

  ADDR18:15 = `1101` multicast to columns 1 + 2 + 4

  etc.

  ADDR18:15 = `1111` multicast to columns 1 + 2 + 3 + 4

- Row/Column Addr

  ADDR22:19 = `0001` – `0011` rows 2, 3, 4, column 1

  ADDR22:19 = `0101` – `0111` rows 2, 3, 4, column 2

  ADDR22:19 = `1001` – `1011` rows 2, 3, 4, column 3

  ADDR22:19 = `1101` – `1111` rows 2, 3, 4, column 4

Note: For broadcast write applications, it is possible to set "SELF_MASK/SELF_ADDR" to non-unique values. This is only allowed if a port is used as a write-only port. This allows multiple UltraRAMs to be written in a single transaction.

The Versal device is supported by the Vivado Design Suite, which includes several code templates to help target the available silicon resources. There are three methods of RTL design entry to use the UltraRAM memories:

- Use the Xilinx Parameterized Macros (XPM)
- Infer an RTL memory and use the ram_style attribute set to "ultra."
- Instantiate the device primitive.

Examples for these methods are in the Vivado language templates accessible from the main Vivado tools menu by selecting **Tools → Language Templates**.

XPM is the most effective method to obtain expected results with a high degree of customization.

Built-in Error Detection and Correction

Each UltraRAM 4K x 72 RAM has built in optional Hamming code error correction for each port. The upper (MSB) 8 bits of the 72-bit data bus are used for parity when ECC is turned on. The ECC operation is transparent to the user. All byte write enable BWE_B[8:0] bits must be set to 1 (High) in ECC mode for proper operation. ECC operations for port A and port B are identical.

Eight protection bits (ECCPARITY) are generated during each write operation and stored with the 64-bit data into the memory. These ECCPARITY bits are used during each read operation to correct any single-bit error, or to detect (but not correct) any double-bit error. ECC data bits and status/control bits are synchronous to the CLK.

During each read operation, 72 bits of data (64 bits of data and 8 bits of parity) are read from the memory and presented to the ECC decoder. The ECC decoder generates two status outputs (SBITERR_A/B and DBITERR_A/B) that are used to indicate the three possible read results: no error, single-bit error corrected, or double-bit error detected. In the standard ECC mode, the read operation does not correct the error in the memory array. It only presents corrected data on DOUT. To improve FMAX, optional registers are available for data output (DO), SBITERR, and DBITERR.

If RST_A/B is asserted, all output registers are reset to 0. Therefore, the SBITERR and DBITERR status signals are also RESET to 0 (Low) indicating that the data output does not have a single bit or a double bit error.

The UltraRAM can also inject errors in either of the ports. ECC mode can inject single bit errors or double bit errors in any or all words. When INJECT_SBITERR is asserted during a write cycle, a single bit error is injected internally in the memory corresponding to DIN[30]. When INJECT_DBITERR is asserted during a write cycle, a double bit error is injected internally in the memory corresponding to DIN[30] and DIN[62]. If both INJECT_SBITERR and INJECT_DBITERR are asserted during a write cycle, a double bit error is injected at the same location as INJECT_DBITERR.

This capability is available in all ECC modes.

ECC Modes

In the standard ECC mode (EN_ECC_RD = TRUE and EN_ECC_WR = TRUE), both encoder and decoder are enabled. During write, 64-bit data and 8-bit ECC generated parity are stored in the array. The external parity input bits are ignored. During read, the 72-bit decoded data and parity are read out.

The most common use case is to enable both the ECC encoder and decoder in a port. However, the encoder and decoder can be enabled separately. To enable only the encoder, the data must be sent through the DI port, the ECCPARITY bits are written into the RAM, and the decoder is disabled. To use only the decoder, the encoder is disabled, the data is written into the RAM, and the corrected data and status bits are read out of the UltraRAM.

ECC Modes of Operation

There are three types of ECC operation:

- Full ECC mode
- ECC ENCODE only mode
- ECC DECODE only mode

The standard ECC mode uses both the encoder and decoder.

Standard ECC

Set by Attributes

EN_ECC_RD = TRUE

EN_ECC_WR = TRUE

Standard ECC Write

The ECC encoder uses DIN[63:0] to generate the corresponding 8 bits of ECC parity, appends it to the 64 data bits, and then writes into the memory. Because ECC parity is generated internally, the DIN[71:64] pins are not used.

The IREG_PRE optional pipeline stage is available before the ECC encode logic for all input pins. This stage can be enabled as needed to meet the maximum frequency requirement.

Standard ECC Read

During read operation, the 72-bit memory content, consisting of 64 bits of data and 8 bits of parity is read out from an address location and decoded internally. If there is no error, the original data and parity are output at DOUT[71:0]. If there is a single-bit error in either the data or the parity, the error is corrected, and SBITERR is High. If there is a double-bit error in the data and parity, the error is not corrected. The original data and parity is output and DBITERR is High.

The OREG optional pipeline stage is available just before the ECC decode logic and the OREG_ECC optional pipeline stage is available just after the ECC decode logic for all the DOUT and error bit outputs. Either or both of these stages can be enabled depending on the maximum frequency and latency requirements of the design.

ECC Encode Only

Set by Attributes

EN_ECC_RD = FALSE

EN_ECC_WR = TRUE

ECC Encode-Only Write

The ECC encoder uses DIN[63:0] to generate the corresponding 8 bits of ECC parity, appends it to the 64 data bits, and then writes into the memory. Because ECC parity is generated internally, the DIN[71:64] pins are not used.

The IREG_PRE optional pipeline stage is available before the ECC encode logic for all input pins. This stage can be enabled as needed to meet the Fmax requirement.

ECC Encode-Only Read

In ECC encode-only mode, read is identical to normal block RAM read. 64-bit data appears at DOUT[63:0] and 8-bit parity appears at DOUT[71:64]. Single-bit error correction does not occur, and the error flags SBITERR and DBITERR are never asserted.

ECC Decode Only

Set by Attributes

EN_ECC_RD = TRUE

EN_ECC_WR = FALSE

In ECC decode-only, only the ECC decoder is enabled. The ECC encoder is disabled. Decode-only mode is used to inject single-bit or double-bit errors to test the functionality of the ECC decoder. The ECC parity bits must be externally supplied using the DIN[71:64] pins.

Using ECC Decode Only to Inject Single-Bit Error

- DIN[71:0] with a single-bit error injected is written into the memory array.
- When the memory location is read out, the data is corrected as needed.
- SBITERR lines up with the corresponding DOUT data.

The ECC decoder also corrects single-bit errors in parity bits.

Using the ECC Decode-Only to Inject Double-Bit Error

- DIN[71:0] with double-bit error injected is written into the memory array.
- When the memory location is accessed, the corrupted data is read out and a double-bit error is detected.
- DBITERR lines up with the corresponding DOUT data.

The ECC decoder also detects when double-bit errors in parity bits occurs, and when a single-bit error in the data bits and a single-bit error in the corresponding parity bits occur.

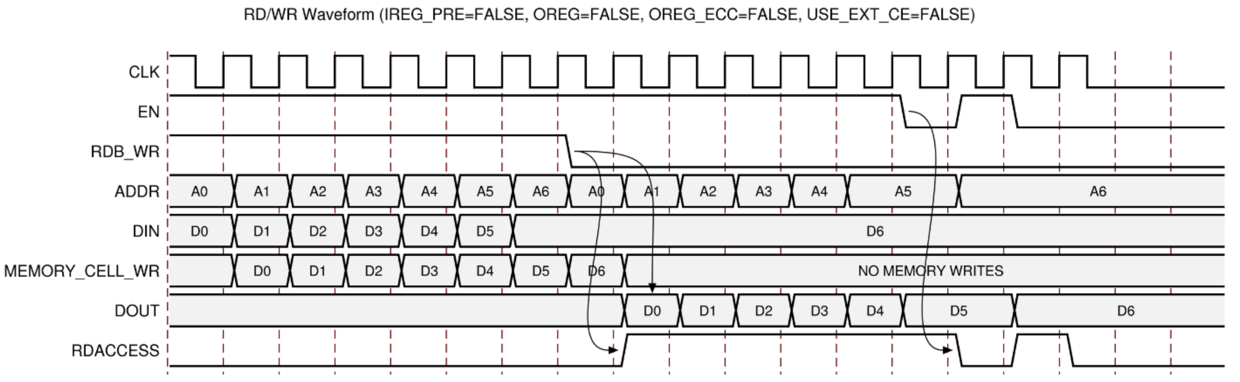
UltraRAM Timing Diagrams

This section describes and illustrates the timing associated with the UltraRAM block. The timing diagrams show the behavior for read/write/reset operations in matrix and single block configuration, as well as the effects of different pipelining options and the clock enable function. Detailed timing diagrams of the sleep and auto sleep modes are shown with various pipelines and latency configurations.

Read/Write Waveforms With and Without Optional Pipeline Registers

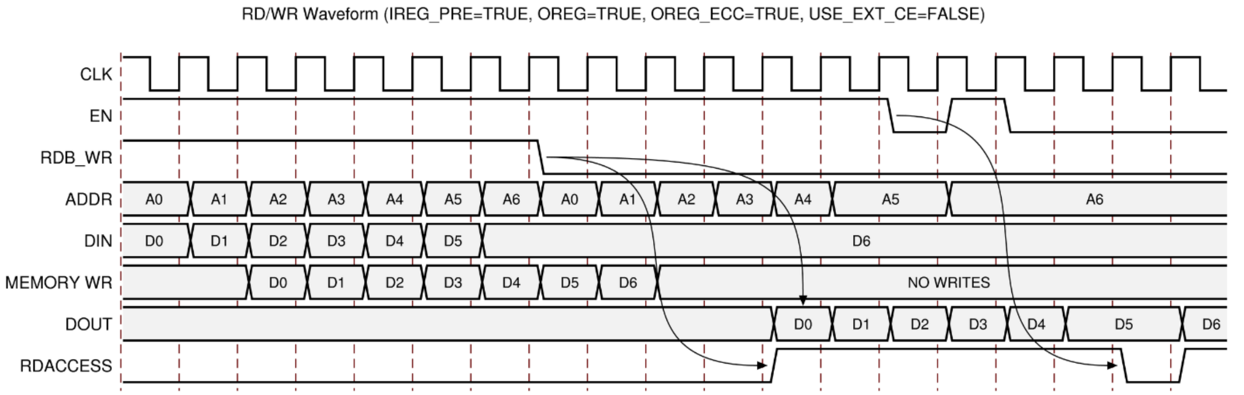
The following figures show the read/write waveforms with and without optional pipeline registers.

Figure 27: **Read/Write with Attributes IREG_PRE_A/B=FALSE, OREG_A/B=FALSE, OREG_ECC_A/B=FALSE, USE_EXT_CE_A/B=FALSE**



X21527-091318

Figure 28: **Read/Write with Attributes IREG_PRE_A/B=TRUE, OREG_A/B=TRUE, OREG_ECC_A/B=TRUE, USE_EXT_CE_A/B=FALSE**


RD/WR Waveform (IREG_PRE=TRUE, OREG=TRUE, OREG_ECC=TRUE, USE_EXT_CE=FALSE)
CLK
EN
RDB_WR
ADDR
A0 A1 A2 A3 A4 A5 A6 A0 A1 A2 A3 A4 A5 A6
DIN
D0 D1 D2 D3 D4 D5 D6
MEMORY WR
D0 D1 D2 D3 D4 D5 D6
NO WRITES
DOUT
D0 D1 D2 D3 D4 D5 D6
RDACCESS

X21528-091318

Read/Write Waveforms With Reset – With and Without Optional Output Pipeline Registers

The following figures show the read/write waveforms with reset and with and without optional pipeline registers.

***Note*:** Reset has priority over the read operation and reset has no impact on any write operation.

Figure 29: **Reset/Read/Write with Attributes RST_MODE=SYNC, IREG_PRE_A/B=FALSE, OREG_A/B=FALSE, OREG_ECC_A/B=FALSE, USE_EXT_CE_A/B=FALSE**

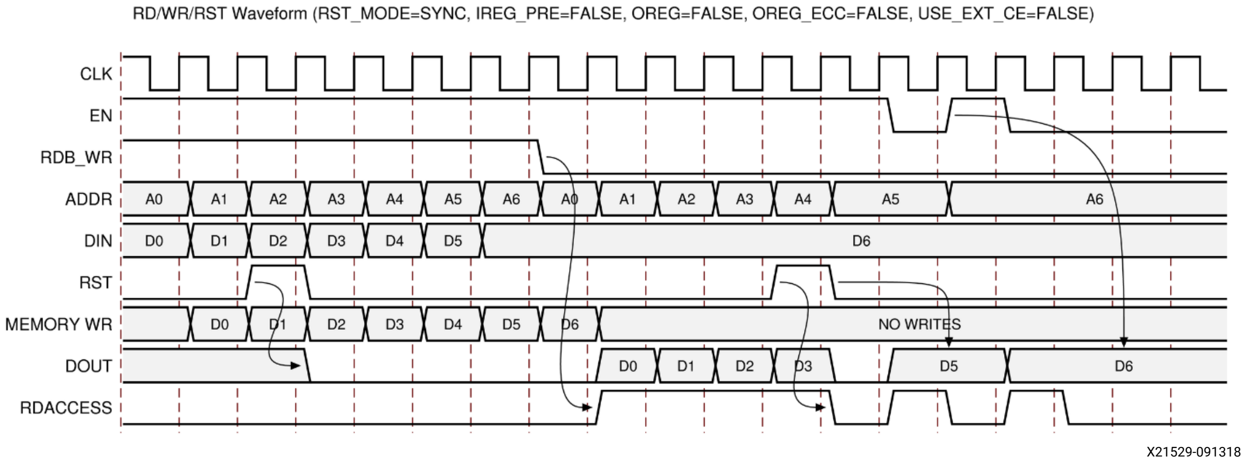

RD/WR/RST Waveform (RST_MODE=SYNC, IREG_PRE=FALSE, OREG=FALSE, OREG_ECC=FALSE, USE_EXT_CE=FALSE)
CLK
EN
RDB_WR
ADDR
A0 A1 A2 A3 A4 A5 A6 A0 A1 A2 A3 A4 A5 A6
DIN
D0 D1 D2 D3 D4 D5 D6
RST
MEMORY WR
D0 D1 D2 D3 D4 D5 D6 NO WRITES
DOUT
D0 D1 D2 D3 D5 D6
RDACCESS
X21529-091318

Figure 30: **Reset/Read/Write with Attributes RST_MODE=SYNC, IREG_PRE_A/B=FALSE, OREG_A/B=TRUE, OREG_ECC_A/B=TRUE, USE_EXT_CE_A/B=FALSE**

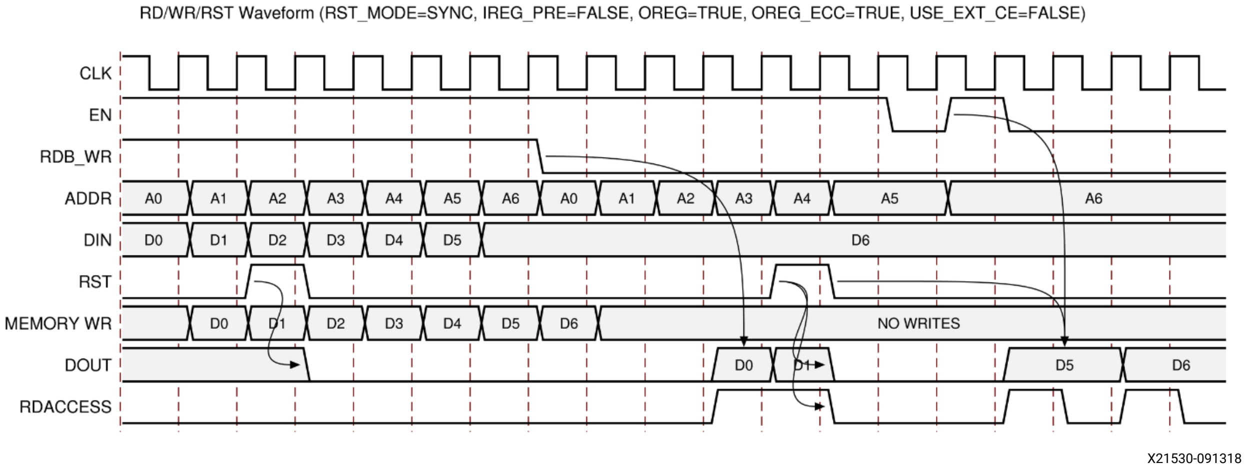
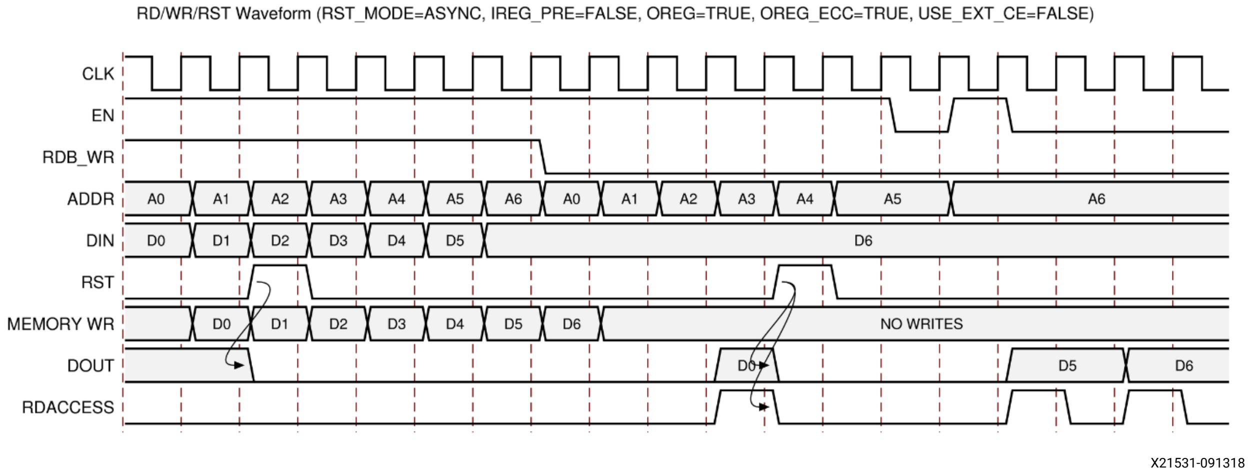

RD/WR/RST Waveform (RST_MODE=SYNC, IREG_PRE=FALSE, OREG=TRUE, OREG_ECC=TRUE, USE_EXT_CE=FALSE)
CLK
EN
RDB_WR
ADDR
A0 A1 A2 A3 A4 A5 A6 A0 A1 A2 A3 A4 A5 A6
DIN
D0 D1 D2 D3 D4 D5 D6
RST
MEMORY WR
D0 D1 D2 D3 D4 D5 D6 NO WRITES
DOUT
D0 D1 D5 D6
RDACCESS
X21530-091318

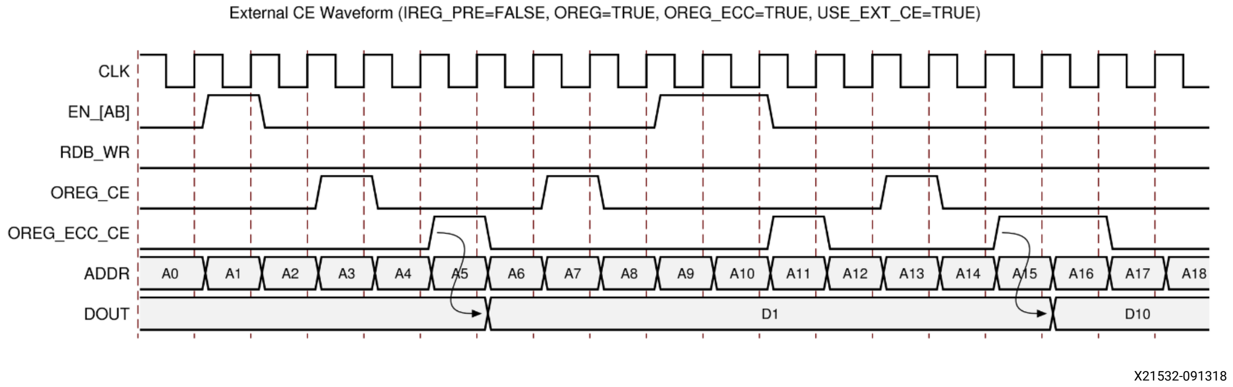
Figure 31: **Read/Write with Attributes RST_MODE=ASYNC, IREG_PRE_A/B=FALSE, OREG_A/B=TRUE, OREG_ECC_A/B=TRUE, USE_EXT_CE_A/B=FALSE**



Read/Write Waveforms With External CE

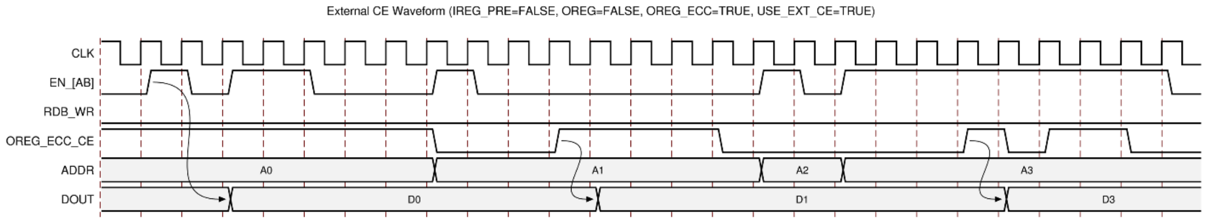
The following figures show read/write waveforms with external CE. Using external CE is allowed for single UltraRAM blocks only, and is not supported for cascaded UltraRAMs. RDACCESS is not supported when USE_EXT_CE=TRUE.

Figure 32: **External CE with Attributes IREG_PRE_A/B=FALSE, OREG_A/B=TRUE, OREG_ECC_A/B=TRUE, USE_EXT_CE_A/B=TRUE**



RDACCESS is not supported when USE_EXT_CE=TRUE.

Figure 33: **External CE with Attributes IREG_PRE_A/B=FALSE, OREG_A/B=FALSE, OREG_ECC_A/B=TRUE, USE_EXT_CE_A/B=TRUE**


External CE Waveform (IREG_PRE=FALSE, OREG=FALSE, OREG_ECC=TRUE, USE_EXT_CE=TRUE)
CLK
EN_[AB]
RDB_WR
OREG_ECC_CE
ADDR
DOUT
A0
A1
A2
A3
D0
D1
D3

X21533-091318

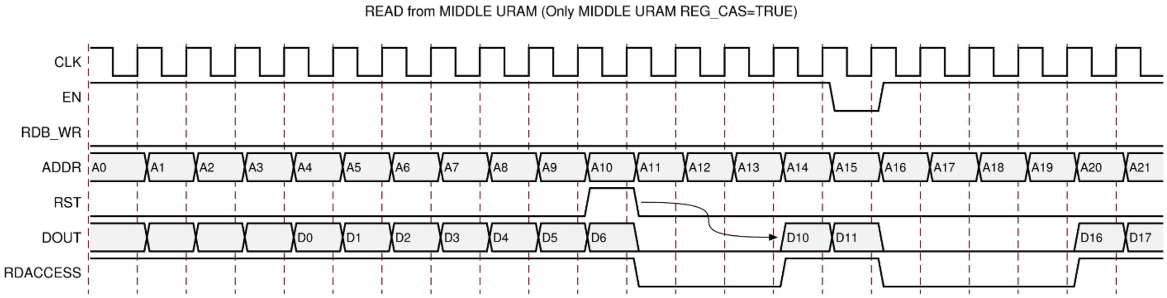
Read From Matrix Waveforms With Reset

The waveforms in the figures are for a three UltraRAM cascade case and assume OREG and OREG_ECC is set to TRUE for all three UltraRAMs in the cascade chain:

- RST input is simultaneously asserted or deasserted at input of all three UltraRAMs.
- All other inputs are driven at input of first UltraRAM. All outputs exit from last UltraRAM.

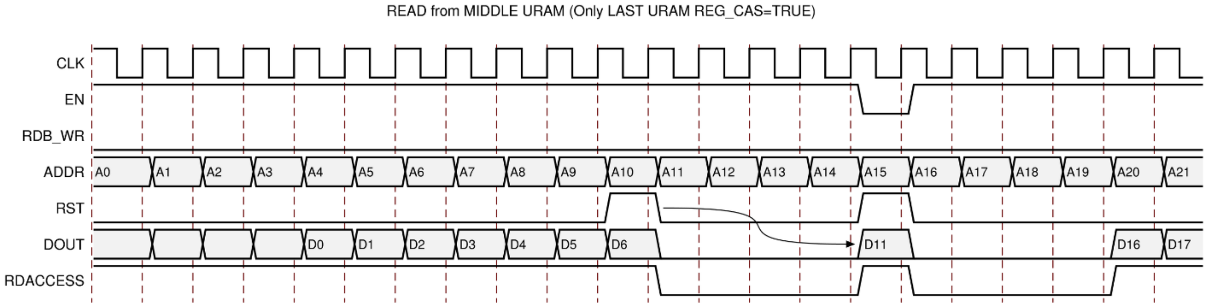
Note**:** The DOUT output behavior after reset might be different depending on the location of REG_CAS, as shown in the figures.

Figure 34: **Reading from Middle UltraRAM – Middle UltraRAM with REG_CAS=TRUE**



X21534-091318

Figure 35: **Reading from Middle UltraRAM – Last UltraRAM with REG_CAS=TRUE**



X21535-091318

Sleep Waveforms

Sleep has priority over EN. Any attempted memory writes are ignored and the previous memory content is preserved. Any attempted memory reads are also ignored. See the following figures.

Figure 36: **Sleep Mode for Write Operations With Attribute IREG_PRE_A/B=FALSE**

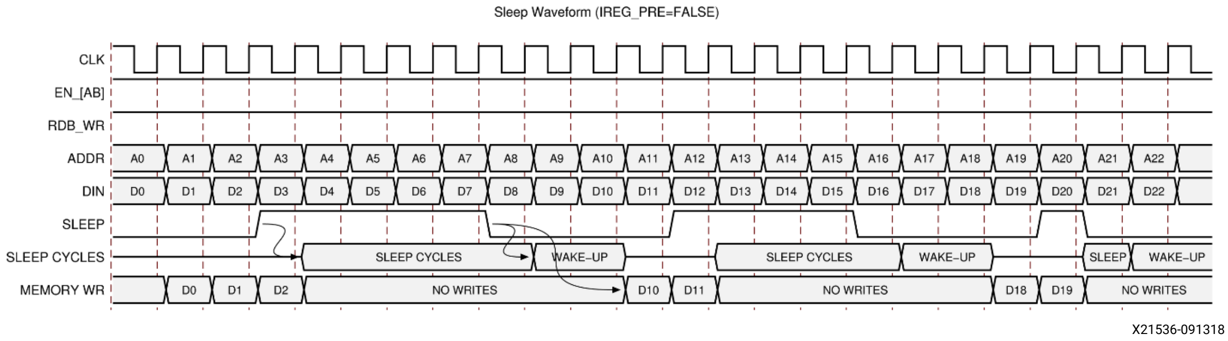

Sleep Waveform (IREG_PRE=FALSE)
CLK
EN_[AB]
RDB_WR
ADDR
A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 A14 A15 A16 A17 A18 A19 A20 A21 A22
DIN
D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 D10 D11 D12 D13 D14 D15 D16 D17 D18 D19 D20 D21 D22
SLEEP
SLEEP CYCLES
SLEEP CYCLES
WAKE–UP
SLEEP CYCLES
WAKE–UP
SLEEP
WAKE–UP
MEMORY WR
D0 D1 D2
NO WRITES
D10 D11
NO WRITES
D18 D19
NO WRITES
X21536-091318

Figure 37: **Sleep Mode for Read Operations With Attributes IREG_PRE_A/B=FALSE, OREG_A/B= FALSE, OREG_ECC_A/B= FALSE, USE_EXT_CE_A/B= FALSE, DOUT is Forced to "0"**

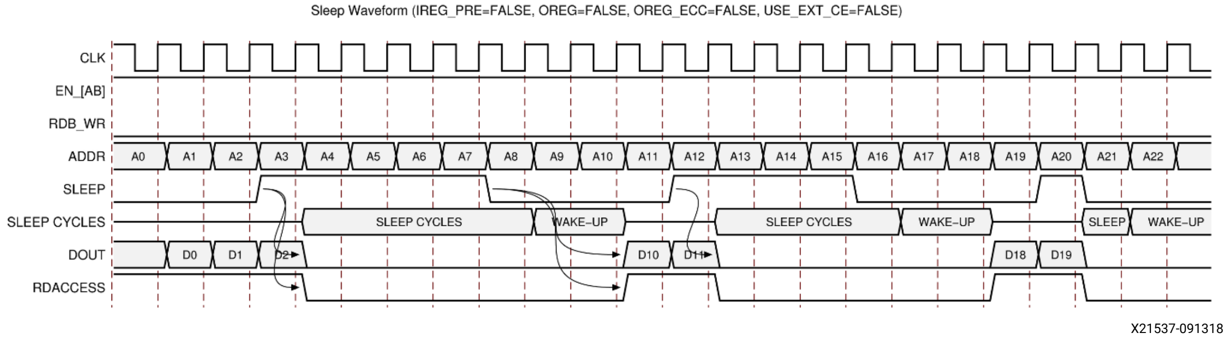

Sleep Waveform (IREG_PRE=FALSE, OREG=FALSE, OREG_ECC=FALSE, USE_EXT_CE=FALSE)
CLK
EN_[AB]
RDB_WR
ADDR
A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 A14 A15 A16 A17 A18 A19 A20 A21 A22
SLEEP
SLEEP CYCLES
SLEEP CYCLES
WAKE–UP
SLEEP CYCLES
WAKE–UP
SLEEP
WAKE–UP
DOUT
D0 D1 D2
D10 D11
D18 D19
RDACCESS
X21537-091318

Figure 38: **Sleep Mode for Read operations With Attributes IREG_PRE_A/B=FALSE, OREG_A/B= FALSE, OREG_ECC_A/B= TRUE, USE_EXT_CE_A/B= FALSE, DOUT is Preserved With Previous Read Data**

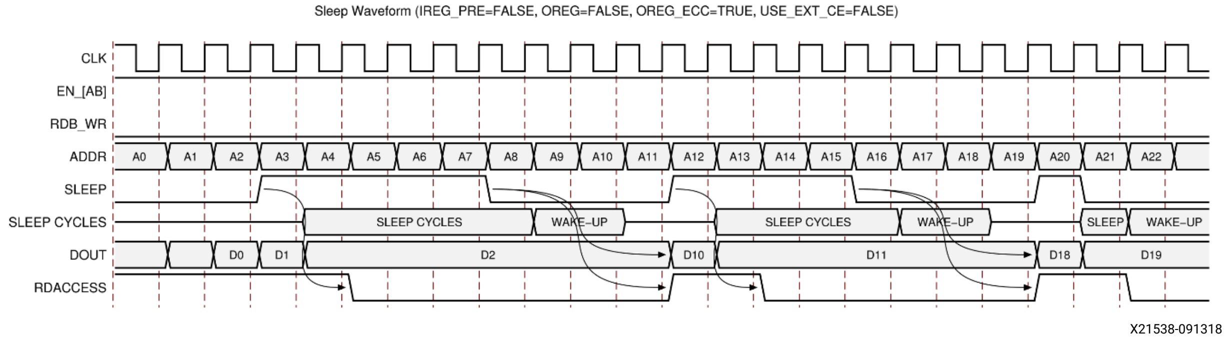

Sleep Waveform (IREG_PRE=FALSE, OREG=FALSE, OREG_ECC=TRUE, USE_EXT_CE=FALSE)
CLK
EN_[AB]
RDB_WR
ADDR
A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 A14 A15 A16 A17 A18 A19 A20 A21 A22
SLEEP
SLEEP CYCLES
SLEEP CYCLES
WAKE–UP
SLEEP CYCLES
WAKE–UP
SLEEP
WAKE–UP
DOUT
D0 D1
D2
D10
D11
D18
D19
RDACCESS
X21538-091318

Figure 39: **Sleep Mode For Read Operations With Attributes IREG_PRE_A/B=FALSE, OREG_A/B= TRUE, OREG_ECC_A/B= FALSE, USE_EXT_CE_A/B= FALSE**

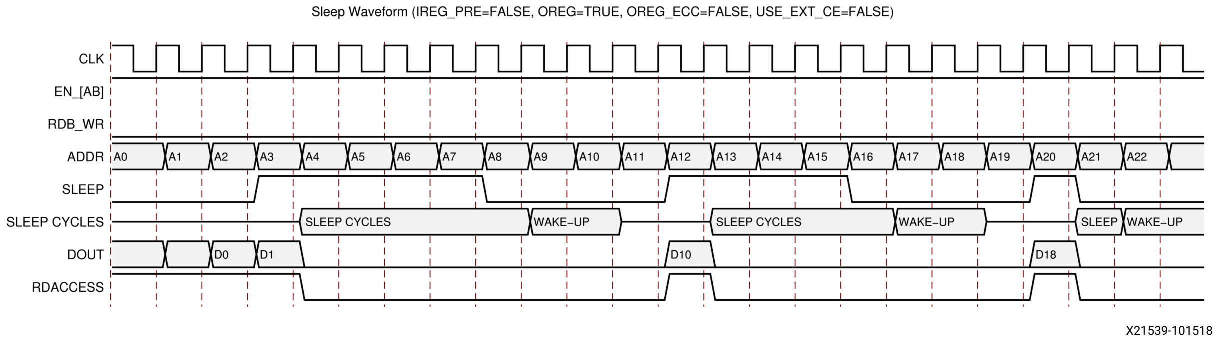
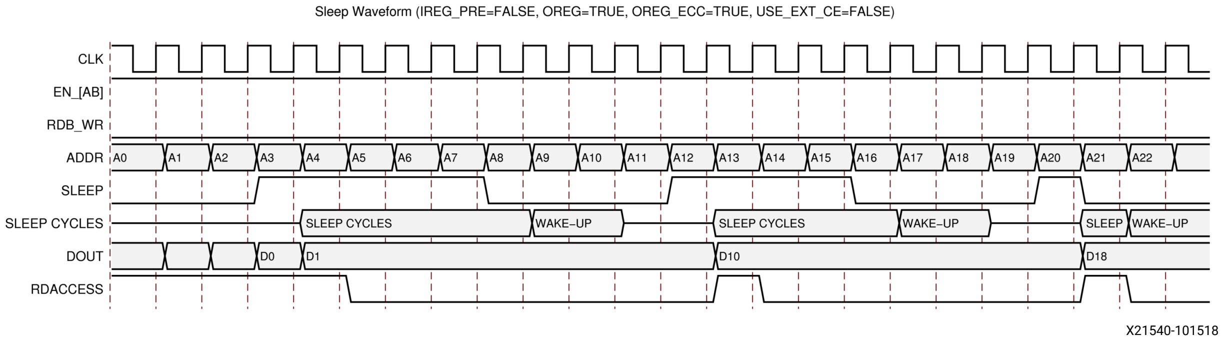

Sleep Waveform (IREG_PRE=FALSE, OREG=TRUE, OREG_ECC=FALSE, USE_EXT_CE=FALSE)
CLK
EN_[AB]
RDB_WR
ADDR
SLEEP
SLEEP CYCLES
DOUT
RDACCESS
A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 A14 A15 A16 A17 A18 A19 A20 A21 A22
SLEEP CYCLES
WAKE–UP
SLEEP CYCLES
WAKE–UP
SLEEP
WAKE–UP
D0
D1
D10
D18
X21539-101518

Figure 40: **Sleep Mode for Read Operations With Attributes IREG_PRE_A/B=FALSE, OREG_A/B= TRUE, OREG_ECC_A/B= TRUE, USE_EXT_CE_A/B= FALSE**


Sleep Waveform (IREG_PRE=FALSE, OREG=TRUE, OREG_ECC=TRUE, USE_EXT_CE=FALSE)
CLK
EN_[AB]
RDB_WR
ADDR
SLEEP
SLEEP CYCLES
DOUT
RDACCESS
A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 A14 A15 A16 A17 A18 A19 A20 A21 A22
SLEEP CYCLES
WAKE–UP
SLEEP CYCLES
WAKE–UP
SLEEP
WAKE–UP
D0
D1
D10
D18
X21540-101518

Read outputs D2, D11, and D19 are lost due to sleep cycle immediately after read (when OREG=TRUE). Consequently, DOUT is driven to "0". However, the RDACCESS is asserted because the internal read memory access is not blocked.

When OREG=TRUE, the read corresponding to address A2 internally is not blocked (because sleep is still Low in this cycle). However, because sleep goes High in the next cycle, the OREG has lost the data, and the output of OREG becomes "0". In this case, even though OREG_ECC=TRUE, because the read itself was not blocked, the OREG_ECC becomes "0" because new read data is expected. This occurs if there is a read operation followed immediately by sleep with OREG=TRUE (irrespective of whether OREG_ECC is TRUE or FALSE).

Auto Sleep Waveforms

To determine when the UltraRAM can activate sleep and wake-up in the auto sleep mode, look-ahead information is required. The byte write enable, read/write, data, and lower address inputs must be delayed with respect to the enable and higher address inputs. These inputs are delayed by pipeline stages equal to the AUTO_SLEEP_LATENCY setting, which can be between 3 and 15. FIFOs or linear shift registers can be used to accomplish this in the ACAP fabric. Other signals, such as INJECT and CE inputs, must also be pipeline aligned if used. See Auto Sleep Latency – AUTO_SLEEP_LATENCY for more information.

Figure 41: **Typical Fabric Implementation for Using Auto Sleep Mode**

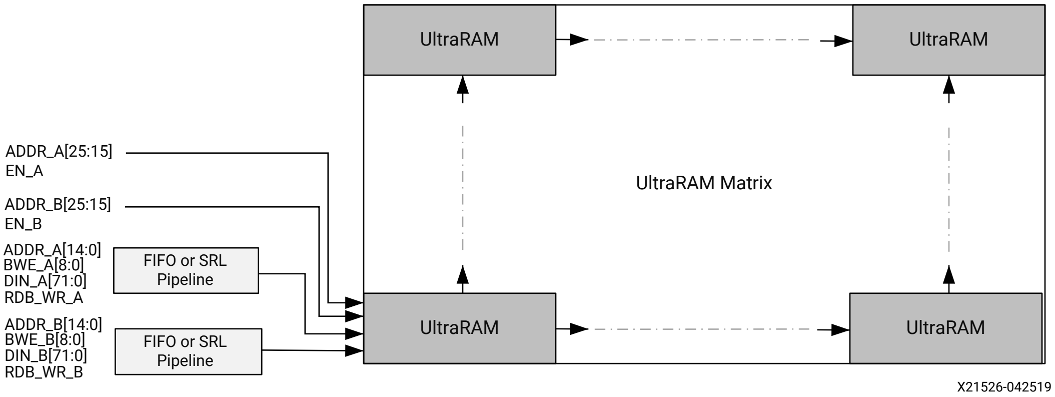
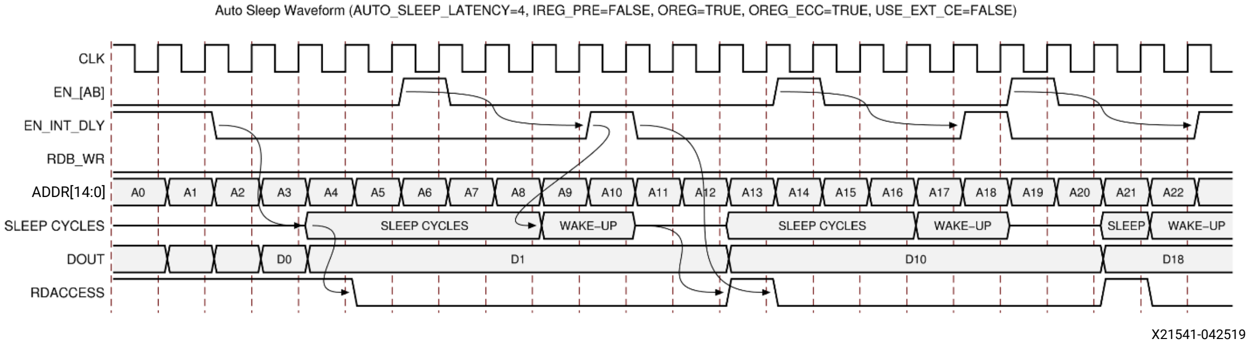


Figure 42: **Auto Sleep Mode for Read Operations With Attributes AUTO_SLEEP_LATENCY=4, IREG_PRE_A/B=FALSE, OREG_A/B= TRUE, OREG_ECC_A/B= TRUE, USE_EXT_CE_A/B= FALSE**

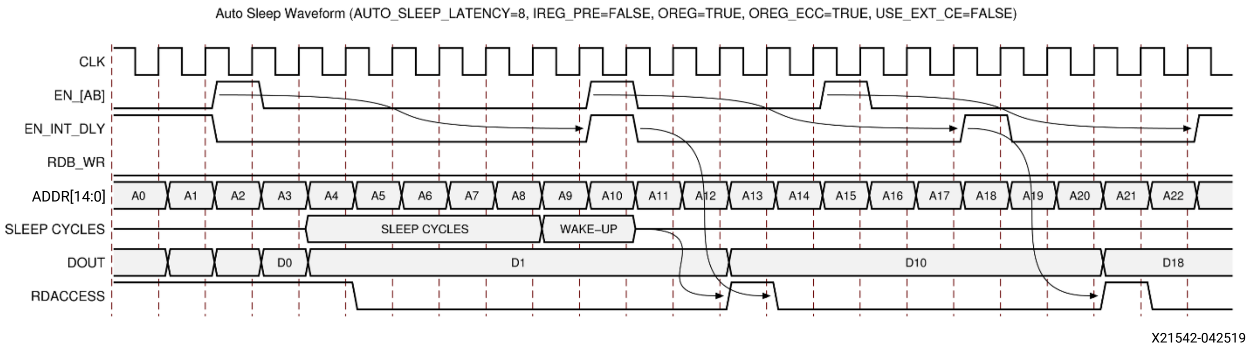


EN_INT_DLY is delayed by AUTO_SLEEP_LATENCY from input EN to show the pipeline alignment with other inputs. DOUT holds previous read data during sleep cycles when OREG_ECC=TRUE.

After EN_INT_DLY is deasserted, an extra idle cycle is inserted before the sleep cycle because of the OREG stage. If the OREG pipeline is set to TRUE, an extra cycle is needed to ensure the read data has propagated to the output before the UltraRAM enters sleep. Consequently, there is a wait of two cycles after EN_INT_DLY deasserts before starting the sleep cycles. Additionally, the sleep cycle is also a function of AUTO_SLEEP_LATENCY, which dictates the number of idle cycles needed to go to sleep (the delay from EN to EN_INT_DLY).

Auto sleep wake-up always occurs one cycle before EN_INT_DLY goes High to guarantee sufficient wake-up time before a next read or write cycle. Wake-up is only a function of EN_INT_DLY_A/B (and EN by extension) rising and no other inputs.

Figure 43: **Auto Sleep Mode for Read Operations With Attributes AUTO_SLEEP_LATENCY=8, IREG_PRE_A/B=FALSE, OREG_A/B= TRUE, OREG_ECC_A/B= TRUE, USE_EXT_CE_A/B= FALSE**

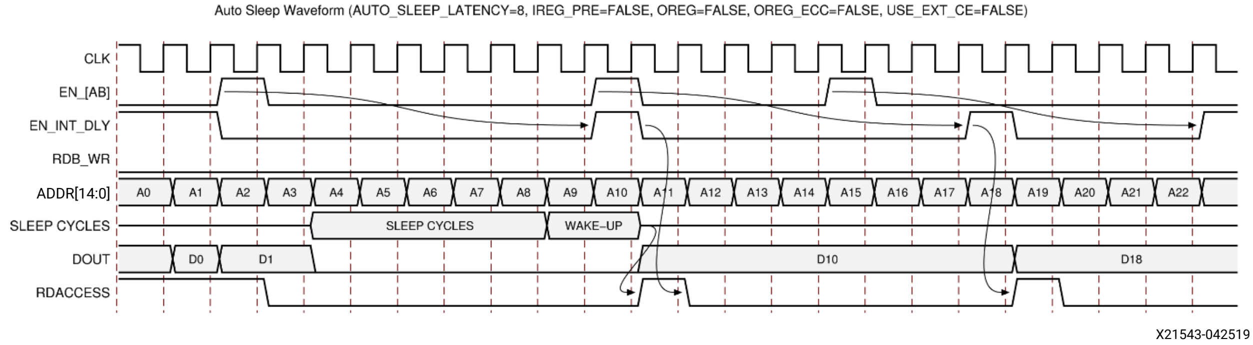


EN_INT_DLY is delayed by AUTO_SLEEP_LATENCY from input EN to show the pipeline alignment with other inputs. DOUT holds previous read data during sleep cycles when OREG_ECC=TRUE.

The AUTO_SLEEP_LATENCY dictates the number of idle cycles needed for the UltraRAM to go to sleep. In the figure above, it is set to eight, which means at least eight cycles are needed between EN_INT_DLY deasserting and the next assertion of EN_INT_DLY to see any sleep cycles. In the later cycles (after sleep/wake-up), there is not enough idle time and the UltraRAM does not go to sleep again.

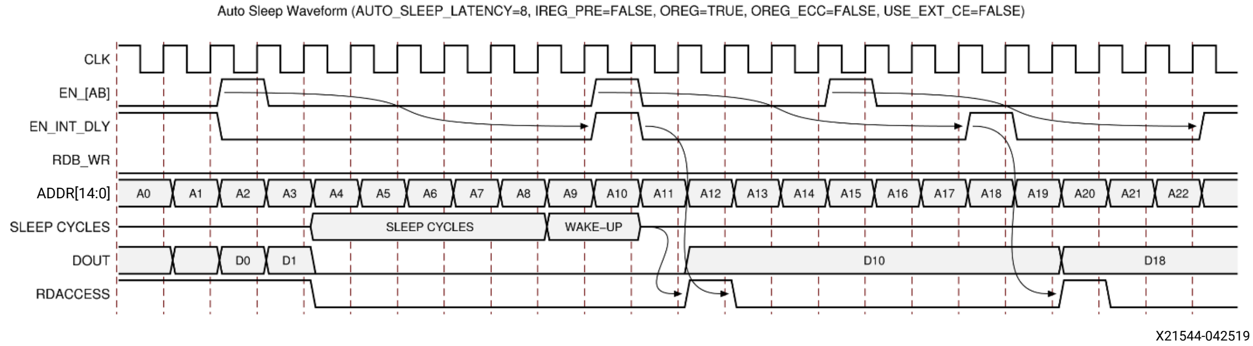
See the attribute description in Auto Sleep Latency – AUTO_SLEEP_LATENCY for the number of expected sleep cycles.

Figure 44: **Auto Sleep Mode for Read Operations With Attributes AUTO_SLEEP_LATENCY=8, IREG_PRE_A/B=FALSE, OREG_A/B= FALSE, OREG_ECC_A/B= FALSE, USE_EXT_CE_A/B= FALSE**



EN_INT_DLY is delayed by AUTO_SLEEP_LATENCY from input EN to show the pipeline alignment with other inputs. DOUT holds previous read data when there are no sleep cycles, but is driven to 0 during sleep cycles when OREG_ECC=FALSE.

Figure 45: **Auto Sleep Mode for Read Operations With Attributes AUTO_SLEEP_LATENCY=8, IREG_PRE_A/B=FALSE, OREG_A/B= TRUE, OREG_ECC_A/B= FALSE, USE_EXT_CE_A/B= FALSE**



EN_INT_DLY is delayed by AUTO_SLEEP_LATENCY from input EN to show the pipeline alignment with other inputs. DOUT holds previous read data when there are no sleep cycles, but is driven to 0 during sleep cycles when OREG_ECC=FALSE.

Additional Resources and Legal Notices

Xilinx Resources

For support resources such as Answers, Documentation, Downloads, and Forums, see Xilinx Support.

Documentation Navigator and Design Hubs

Xilinx® Documentation Navigator (DocNav) provides access to Xilinx documents, videos, and support resources, which you can filter and search to find information. To open DocNav:

- From the Vivado® IDE, select **Help → Documentation and Tutorials**.
- On Windows, select **Start → All Programs → Xilinx Design Tools → DocNav**.
- At the Linux command prompt, enter `docnav`.

Xilinx Design Hubs provide links to documentation organized by design tasks and other topics, which you can use to learn key concepts and address frequently asked questions. To access the Design Hubs:

- In DocNav, click the **Design Hubs View** tab.
- On the Xilinx website, see the Design Hubs page.

***Note*:** For more information on DocNav, see the Documentation Navigator page on the Xilinx website.

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 https://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 https://www.xilinx.com/legal.htm#tos.

This document contains preliminary information and is subject to change without notice. Information provided herein relates to products and/or services not yet available for sale, and provided solely for information purposes and are not intended, or to be construed, as an offer for sale or an attempted commercialization of the products and/or services referred to herein.

AUTOMOTIVE APPLICATIONS DISCLAIMER

AUTOMOTIVE PRODUCTS (IDENTIFIED AS "XA" IN THE PART NUMBER) ARE NOT WARRANTED FOR USE IN THE DEPLOYMENT OF AIRBAGS OR FOR USE IN APPLICATIONS THAT AFFECT CONTROL OF A VEHICLE ("SAFETY APPLICATION") UNLESS THERE IS A SAFETY CONCEPT OR REDUNDANCY FEATURE CONSISTENT WITH THE ISO 26262 AUTOMOTIVE SAFETY STANDARD ("SAFETY DESIGN"). CUSTOMER SHALL, PRIOR TO USING OR DISTRIBUTING ANY SYSTEMS THAT INCORPORATE PRODUCTS, THOROUGHLY TEST SUCH SYSTEMS FOR SAFETY PURPOSES. USE OF PRODUCTS IN A SAFETY APPLICATION WITHOUT A SAFETY DESIGN IS FULLY AT THE RISK OF CUSTOMER, SUBJECT ONLY TO APPLICABLE LAWS AND REGULATIONS GOVERNING LIMITATIONS ON PRODUCT LIABILITY.

Copyright