Spartan-6 FPGA Block RAM Resources

User Guide

UG383 (v1.2) February 23, 2010

XILINX®

Xilinx is disclosing this user guide, manual, release note, and/or specification (the "Documentation") to you solely for use in the development of designs to operate with Xilinx hardware devices. You may not reproduce, distribute, republish, download, display, post, or transmit the Documentation in any form or by any means including, but not limited to, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of Xilinx. Xilinx expressly disclaims any liability arising out of your use of the Documentation. Xilinx reserves the right, at its sole discretion, to change the Documentation without notice at any time. Xilinx assumes no obligation to correct any errors contained in the Documentation, or to advise you of any corrections or updates. Xilinx expressly disclaims any liability in connection with technical support or assistance that may be provided to you in connection with the Information.

THE DOCUMENTATION IS DISCLOSED TO YOU “AS-IS” WITH NO WARRANTY OF ANY KIND. XILINX MAKES NO OTHER WARRANTIES, WHETHER EXPRESS, IMPLIED, OR STATUTORY, REGARDING THE DOCUMENTATION, INCLUDING ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT OF THIRD-PARTY RIGHTS. IN NO EVENT WILL XILINX BE LIABLE FOR ANY CONSEQUENTIAL, INDIRECT, EXEMPLARY, SPECIAL, OR INCIDENTAL DAMAGES, INCLUDING ANY LOSS OF DATA OR LOST PROFITS, ARISING FROM YOUR USE OF THE DOCUMENTATION.

© 2009–2010 Xilinx, Inc. XILINX, the Xilinx logo, Virtex, Spartan, ISE, and other designated brands included herein are trademarks of Xilinx in the United States and other countries. All other trademarks are the property of their respective owners.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
06/24/09	1.0	Initial Xilinx release.
10/29/09	1.1	Added last bullet to Asynchronous Clocking discussion. Revised the Unused Inputs connection to Low.
02/23/10	1.2	Changed the Possible Configurations section which includes removal of the 9 Kb block RAM (simple dual-port operation) section. Additional changes throughout the document involve removal of this information. Added port names to Table 2 through Table 3. Updated Asynchronous Clocking discussion. Changed Data and Address Width - DATA_WIDTH_A, DATA_WIDTH_B. Updated Figure 14.

Table of Contents

Revision History 2

Preface: About This Guide

Additional Documentation 5
Additional Resources 6

Block RAM Resources

Summary 7
Possible Configurations 7
9 Kb Block RAM—Simple Dual-Port Operation 8
9 Kb Block RAM—True Dual-Port Operation 8
18 Kb Block RAM—True Dual-Port Operation 9
Spartan-6 FPGA Block RAM Features 11
Spartan-6 FPGA Block RAM Usage Rules 11
Synchronous Dual-Port and Simple Dual-Port RAMs 11
Data Flow 11
Read Operation 13
Write Operation 13
Write Modes 13
WRITE_FIRST or Transparent Mode (Default) 13
READ_FIRST or Read-Before-Write Mode 14
NO_CHANGE Mode 14
Conflict Avoidance 15
Asynchronous Clocking 15
Synchronous Clocking 15
Optional Output Registers 15
Simple Dual-Port Block RAM 16
Byte-wide Write Enable 17
Block RAM Library Primitives 18
Block RAM Port Signals 19
Clock - CLKA, CLKB (CLKAWRCLK, CLKBRDCLK) 19
Enable - ENA, ENB (ENAWREN, ENBRDEN) 20
Byte-wide Write Enable - WEA, WEB (WEAWEL, WEBWEU) 20
Register Enable - REGCEA, REGCEB (REGCEA, REGCEBREGCE) 20
Set/Reset - RSTA, RSTB (RSTA, RSTBRST) 20
Address Bus - ADDRA, ADDRB (ADDRAWRADDR, ADDRBRDADDR) 21
Data-In Buses - DIA, DIB (DIADI, DIBDI) 21
Data-In Parity - DIPA, DIPB (DIPADIP, DIPBDIP) 21
Data-Out Buses - DOA, DOB (DOADO, DOBDO) 21
Data-Out Parity - DOPA, DOPB (DOPADOP, DOPBDOP) 21
GSR 22
Unused Inputs 22
Block RAM Address Mapping 22
Block RAM Attributes 22
Data and Address Width - DATA_WIDTH_A, DATA_WIDTH_B 22

Content Initialization - INIT_xx 23
Content Initialization - INITP_xx 24
Output Latches Initialization - INIT_A, INIT_B 24
Output Latches/Registers Synchronous Set/Reset - SRVAL_A, SRVAL_B 24
RAM MODE - RAM_MODE 24
Reset or CE Priority - RST_PRIORITY_A, RST_PRIORITY_B 24
Data Latch Reset - EN_RSTRAM_A, EN_RSTRAM_B 24
Reset Type - RSTTYPE 25
Optional Output Register On/Off Switch - DO[A|B]_REG 25
Write Mode - WRITE MODE_A, WRITE_MODE_B 25
Block RAM Location Constraints 25
Block RAM Initialization 25
Additional Block RAM Primitive Design Considerations 25
Optional Output Registers 25
RAMB16BWER and RAMB8BWER Port Mapping Design Rules 26
Reset Behavior 26
Byte-wide Write Enable 26
Block RAM Applications 26
Creating Larger RAM Structures 26
Block RAM RST in Register Mode 27
Block RAM Timing Model 28
Block RAM Timing Parameters 28
Block RAM Timing Characteristics 29
Clock Event 1 30
Clock Event 2 31
Clock Event 4 31
Clock Event 5 31
Block RAM Timing Model 31

About This Guide

This guide serves as a technical reference describing the Spartan®-6 FPGA block RAMs available in all Spartan-6 FPGAs. Block RAMs are used for efficient data storage or buffering, for high-performance state machines or FIFO buffer, for large shift registers, large look-up tables, or ROMs.

Additional Documentation

The following documents are also available for download at http://www.xilinx.com/support/documentation/spartan-6.htm.

- Spartan-6 Family Overview

  This overview outlines the features and product selection of the Spartan-6 family.

- Spartan-6 FPGA Data Sheet: DC and Switching Characteristics

  This data sheet contains the DC and switching characteristic specifications for the Spartan-6 family.

- Spartan-6 FPGA Packaging and Pinout Specifications

  This specification includes the tables for device/package combinations and maximum I/Os, pin definitions, pinout tables, pinout diagrams, mechanical drawings, and thermal specifications.

- Spartan-6 FPGA Configuration User Guide

  This all-encompassing configuration guide includes chapters on configuration interfaces (serial and parallel), multi-bitstream management, bitstream encryption, boundary-scan and JTAG configuration, and reconfiguration techniques.

- Spartan-6 FPGA SelectIO Resources User Guide

  This guide describes the SelectIO™ resources available in all Spartan-6 devices.

- Spartan-6 FPGA Clocking Resources User Guide

  This guide describes the clocking resources available in all Spartan-6 devices, including the DCMs and PLLs.

- Spartan-6 FPGA Configurable Logic Block User Guide

  This guide describes the capabilities of the configurable logic blocks (CLBs) available in all Spartan-6 devices.

- Spartan-6 FPGA GTP Transceivers User Guide

  This guide describes the GTP transceivers available in the Spartan-6 LXT FPGAs.

- Spartan-6 FPGA DSP48A1 Slice User Guide

  This guide describes the architecture of the DSP48A1 slice in Spartan-6 FPGAs and provides configuration examples.

- Spartan-6 FPGA Memory Controller User Guide

  This guide describes the Spartan-6 FPGA memory controller block, a dedicated embedded multi-port memory controller that greatly simplifies interfacing Spartan-6 FPGAs to the most popular memory standards.

- Spartan-6 FPGA PCB Design Guide

  This guide provides information on PCB design for Spartan-6 devices, with a focus on strategies for making design decisions at the PCB and interface level.

Additional Resources

To search the Answer Database of silicon, software, and IP questions and answers, or to create a technical support WebCase, see the Xilinx website at:

http://www.xilinx.com/support.

Block RAM Resources

Summary

The block RAM in Spartan-6 FPGAs stores up to 18K bits of data and can be configured as either two independent 9 Kb RAMs, or one 18 Kb RAM. Each RAM can be addressed through two ports, but can also be configured as a single-port RAM. The block RAM resources include output registers to increase pipeline performance. Block RAMs are placed in columns. The total number of block RAMs depends on the size of the Spartan-6 device.

Similar to other Xilinx 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. 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 either remain unchanged, reflect the new data being written or the previous data now being overwritten.

Embedded dual- or single-port RAM modules, ROM modules, synchronous FIFOs, and data-width converters are easily implemented using the Xilinx CORE Generator™ block memory modules. Dual-clock FIFOs can be generated using the CORE Generator FIFO Generator module.

Possible Configurations

In true dual-port mode, either port can be a Write or a Read port, independent of the function of the other port. The 9 Kb RAM has a maximum of 18-bit data width unless operating in simple dual-port mode. In simple dual-port mode, port A can only write and port B can only read, however, the data bus is expanded to 36 bits.

The width of the two ports can be configured independently for true dual-port mode in both block RAM densities. The 9 Kb RAM, configured in simple dual-port mode only, functions in the full x36 data-width configuration, as described in Table 1 through Table 3.

9 Kb Block RAM—Simple Dual-Port Operation

The allowed port-width combinations in simple dual-port mode are shown in Table 1.

Table 1: **Simple Dual-Port Mode Allowed Combinations for 9 Kb Block RAM**

<table>
<tr><th colspan="3" rowspan="3"></th><th colspan="9">Port A</th></tr>
<tr><th colspan="6">No Parity Bits</th><th colspan="3">With Parity Bits</th></tr>
<tr><th>8K x 1</th><th>4K x 2</th><th>2K x 4</th><th>1K x 8</th><th>512 x 16</th><th>256 x 32</th><th>1K x 9</th><th>512 x 18</th><th>256 x 36</th></tr>
<tr><td rowspan="9">Port B</td><td rowspan="6">No parity bits</td><td>8K x 1</td><td colspan="5" rowspan="5">Use true dual-port mode</td><td rowspan="5">None Allowed</td><td colspan="3" rowspan="6">None Allowed</td></tr>
<tr><td>4K x 2</td></tr>
<tr><td>2K x 4</td></tr>
<tr><td>1K x 8</td></tr>
<tr><td>512 x 16</td></tr>
<tr><td>256 x 32</td><td colspan="5">None Allowed</td><td>Allowed</td></tr>
<tr><td rowspan="3">With parity bits</td><td>1K x 9</td><td colspan="6" rowspan="3">None Allowed</td><td colspan="2">Use true dual-port mode</td><td>None Allowed</td></tr>
<tr><td>512 x 18</td></tr>
<tr><td>256 x 36</td><td colspan="2">None Allowed</td><td>Allowed</td></tr>
</table>

9 Kb Block RAM—True Dual-Port Operation

The allowed port-width combinations in true dual-port mode are shown in Table 2.

Table 2: **True Dual-Port Mode Allowed Combinations for 9 Kb Block RAM**

<table>
<tr><th colspan="2" rowspan="3"></th><th colspan="7">Port A</th></tr>
<tr><th colspan="5">No Parity Bits</th><th colspan="2">With Parity Bits</th></tr>
<tr><th>8K x 1</th><th>4K x 2</th><th>2K x 4</th><th>1K x 8</th><th>512 x 16</th><th>1K x 9</th><th>512 x 18</th></tr>
<tr><td rowspan="6">Port B</td><td>8K x 1</td><td colspan="5" rowspan="5">All Allowed</td><td colspan="2" rowspan="5">None Allowed</td></tr>
<tr><td>4K x 2</td></tr>
<tr><td>2K x 4</td></tr>
<tr><td>1K x 8</td></tr>
<tr><td>512 x 16</td></tr>
<tr><td>1K x 9</td><td colspan="5" rowspan="2">None Allowed</td><td colspan="2" rowspan="2">All Allowed</td></tr>
<tr><td>512 x 18</td></tr>
</table>

18 Kb Block RAM—True Dual-Port Operation

The allowed port-width combinations in true dual-port mode are shown in Table 3.

Table 3: **True Dual-Port Mode Allowed Combinations for 18 Kb Block RAM**

		Port A								
		No Parity Bits						With Parity Bits		
		16K x 1	8K x 2	4K x 4	2K x 8	1K x 16	512 x 32	2K x 9	1K x 18	512 x 36
Port B	16K x 1	All Allowed						None Allowed		
	8K x 2									
	4K x 4									
	2K x 8									
	1K x 16									
	512 x 32									
	2K x 9	None Allowed						All Allowed		
	1K x 18									
	512 x 36									

Table 4 lists an additional way to evaluate the 9 Kb and 18 Kb block RAM combination choices. In addition, the ADDR port widths are listed.

Table 4: **Block RAM Data Combinations and ADDR Locations**

Combinations	Memory Depth	Data Width	Parity Width	Data Input Data Output	ADDR	Total RAM (Kb)
9 Kb Block RAM With and Without Parity						
256 x 32(1)	256	32	NA	[31:0]	[12:5]	8
256 x 36(1)	256	32	4	[35:0]	[12:5]	9
512 x16	512	16	NA	[15:0]	[12:4]	8
512 x18	512	16	2	[17:0]	[12:4]	9
1K x 8	1024	8	NA	[7:0]	[12:3]	8
1K x 9	1024	8	1	[8:0]	[12:3]	9
2K x 4	2048	4	NA	[3:0]	[12:2]	8
4K x 2	4096	2	NA	[1:0]	[12:1]	8
8K x 1	8192	1	NA	[0:0]	[12:0]	8
18 Kb Block RAM With and Without Parity						
512 x 32	512	32	NA	[31:0]	[13:5]	16
512 x 36	512	32	4	[35:0]	[13:5]	18
1K x16	1024	16	NA	[15:0]	[13:4]	16
1K x18	1024	16	2	[17:0]	[13:4]	18
2K x 8	2045	8	NA	[7:0]	[13:3]	16
2K x 9	2048	8	1	[8:0]	[13:3]	18
4K x 4	4096	4	NA	[3:0]	[13:2]	16
8K x 2	8192	2	NA	[1:0]	[13:1]	16
16K x 1	16384	1	NA	[0:0]	[13:0]	16

Notes:

1. x32 and x36 data widths available in simple dual-port (SDP) mode only.

Spartan-6 FPGA Block RAM Features

- Per block memory storage capability where each block RAM can store up to 18 Kb of data.
- Support of two independent 9 Kb blocks, or a single 18 Kb block RAM.
- Each 9 Kb block RAM can be set to simple dual-port mode, doubling data width of the block RAM to a maximum of 36 bits. Simple dual-port mode is defined as having one read-only port and one write-only port with independent clocks.
- Separate synchronous Set/Reset pins to independently control the Set/Reset of the optional output registers and output latch stages in the block RAM. This better harmonizes the Spartan-6 FPGA block RAM with other FPGA families and eases mapping of FPGA logic registers into the block RAM blocks.
- 18 or 36-bit wide ports can have an individual write enable per byte. This feature is popular for interfacing to an on-chip 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. The state of the output port does not change until the port executes another read or write operation. The default block RAM output is latch mode.
- The output data path has an optional internal pipeline register. Using the register mode is strongly recommended. This allows a higher clock rate, however, it adds a clock cycle latency of one.

Spartan-6 FPGA Block RAM Usage Rules

- The block RAM synchronous output registers (optional) are set or reset (SRVAL) with RST when DO_REG = 1. The RST_PRIORITY attribute determines if RST has priority over REGCE. The synchronous output latches are set or reset (SRVAL) with RST when EN_RSTRAM = TRUE. The block RAM must be enabled (EN = 1) for this function.
- The setup time of the block RAM address and write enable pins must not be violated. Violating the address setup time (even if write enable is Low) will corrupt the data contents of the block RAM.

Synchronous Dual-Port and Simple Dual-Port RAMs

Data Flow

The true dual-port 18 Kb block RAM dual-port memories consist of a 18 Kb storage area and two completely independent access ports, A and B. Similarly, each 9 Kb block RAM dual-port memory consists of an 9 Kb storage area and two completely independent access ports, A and B. The structure is fully symmetrical, and both ports are interchangeable. Figure 1 illustrates the true dual-port data flow. Table 5 lists the port names 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, 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. It is up to the user to time the two clocks appropriately. Conflicting simultaneous writes to the same location never cause any physical damage but can result in data uncertainty.

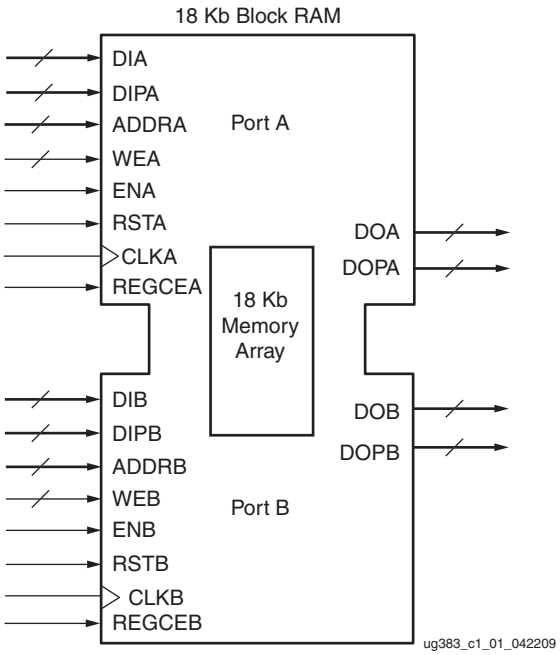


Figure 1: **True Dual-Port Data Flows**

Table 5: **True Dual-Port Names and Descriptions**

Port Name	Description
DI[A | B](1)	Data Input Bus
DIP[A | B](1)	Data Input Parity Bus
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.
RST[A | B]	Synchronous Set/Reset the output registers (DO_REG = 1).
CLK[A | B]	Clock Input
DO[A | B](1)	Data Output Bus
DOP[A | B](1)	Data Output Parity Bus
REGCE[A | B]	Output Register Clock Enable

Notes:

1. The Data-In Buses - DIA, DIB (DIADI, DIBDI) section has more information on data parity pins.

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 will take one extra latency cycle to arrive at the output.

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 determines the behavior of the data available on the output latches after a write clock edge: WRITE_FIRST, READ_FIRST, and NO_CHANGE. The Write mode attribute can be individually selected for each port. The default mode is WRITE_FIRST. 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 (Default)

In WRITE_FIRST mode, the input data is simultaneously written into memory and stored in the data output (transparent write), as shown in Figure 2. These waveforms correspond to latch mode when the optional output pipeline register is not used.

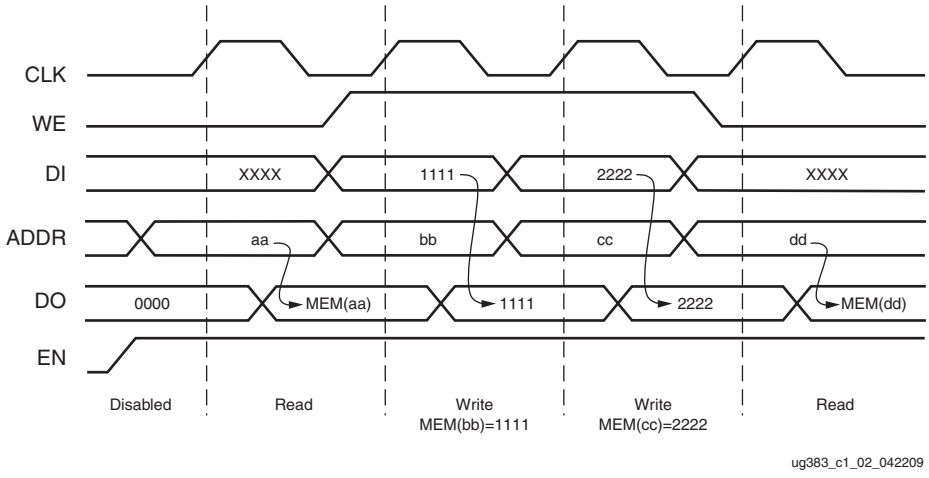


Figure 2: **WRITE_FIRST Mode Waveforms**

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 Figure 3 correspond to latch mode when the optional output pipeline register is not used.

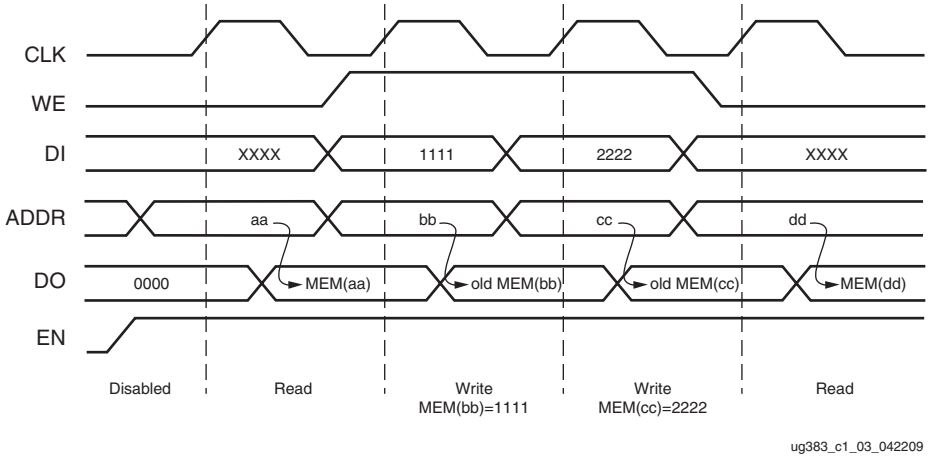


Figure 3: **READ_FIRST Mode Waveforms**

NO_CHANGE Mode

In NO_CHANGE mode, the output latches remain unchanged during a write operation. As shown in Figure 4, 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.

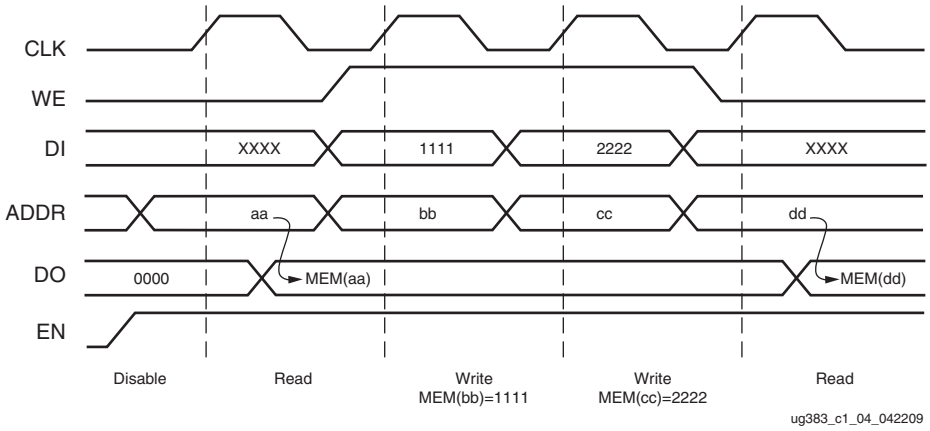


Figure 4: **NO_CHANGE Mode Waveforms**

Conflict Avoidance

Spartan-6 FPGA block RAM memory is a true dual-port RAM where both ports can access any memory location at any time. When accessing the same memory location from both ports, the user must, however, observe certain restrictions. There are two fundamentally different situations: The two ports either have a common clock (synchronous clocking), or the clock frequency and phase is different for the two ports (asynchronous clocking).

Asynchronous Clocking

Asynchronous clocking is the more general case, where the active edges of both clocks do not occur simultaneously:

- There are no timing restrictions when both ports perform a read operation.
- When one port performs a write operation, the other port must not read- or write-access the same memory location. The simulation model will produce an error if this condition is violated. If this restriction is ignored, a read or write operation will produce unpredictable results. There is, however, no risk of physical damage to the device. If a read and write operation is performed, then the write will store valid data at the write location.
- In READ_FIRST mode only, the dual-port block RAM has the additional restriction that addresses for port A and B cannot collide. This applies for both TDP and SDP modes. A read/write on one port and a write operation from the other port at the same address is not allowed. This restriction on the operation of the block RAM should not be ignored.
  - RAMB16BWER when both ports are 18 bits wide or smaller: A13–A6, including A4, cannot be the same.
  - RAMB16BWER when any one port is 36 bits wide: A13–A7, including A5, cannot be the same.
  - RAMB8BWER in all configurations: A12–A6 including A4 cannot be the same.

Synchronous Clocking

Synchronous clocking is the special case, where the active edges of both port clocks occur simultaneously:

- There are no timing restrictions when both ports perform a read operation.
- When one port performs a write operation, the other port must not write into the same location, unless both ports write identical data.
- When one port performs a write operation, the write operation succeeds; the other port can reliably read data from the same location if the write port is in READ_FIRST mode. DATA_OUT on both ports will then reflect the previously stored data.

  If the write port is in either WRITE_FIRST or in NO_CHANGE mode, then the DATA_OUT on the read port would become invalid (unreliable). The mode setting of the read-port does not affect this operation.

Optional Output Registers

The optional output registers improve design performance by eliminating routing delay to the CLB flip-flops for pipelined operation. An independent 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. Figure 5 shows the optional output register.

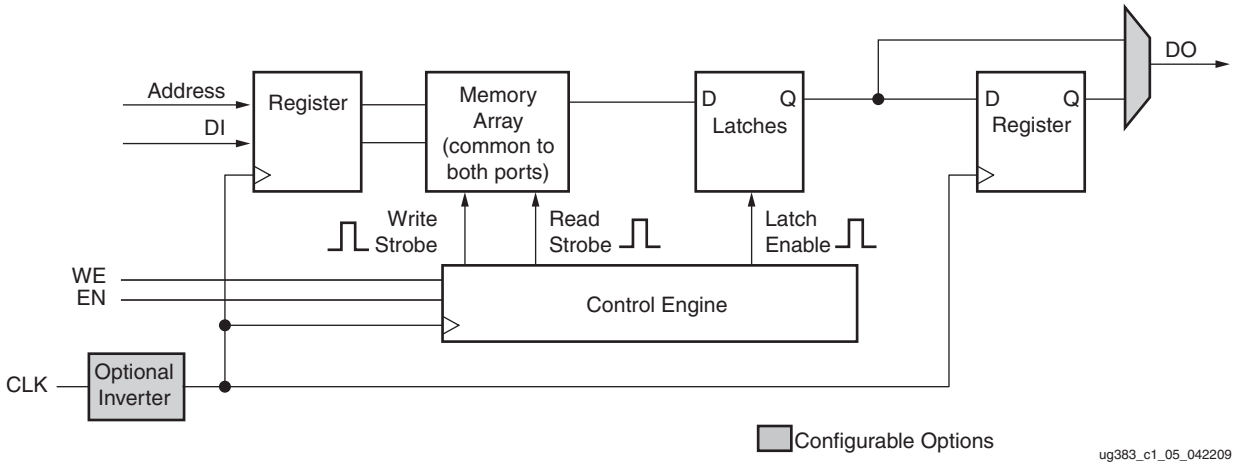


Figure 5: **Block RAM Logic Diagram (One Port Shown)**

Simple Dual-Port Block RAM

Each 9 Kb block can also be configured in a simple dual-port (SDP) RAM mode. In this mode, the block RAM port width doubles to 36 bits for the 9 Kb block RAM. In simple dual-port mode, independent Read and Write operations can occur simultaneously, where port A is the primary Write port and port B is the primary Read port. When the Read and Write port access the same data location at the same time, it is treated as a collision, similar to the port collision in true dual-port mode. Readback through the configuration port is not supported in simple dual-port block RAM mode. Figure 6 shows the simple dual-port data flow. Table 6 lists the SDP port names and descriptions.

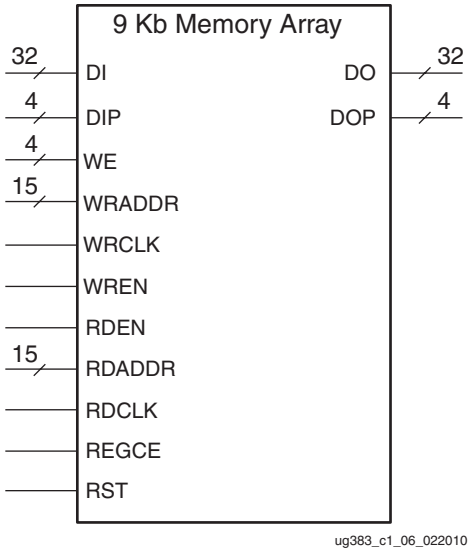


Figure 6: **Simple Dual-Port Data Flow**

Table 6: **Simple Dual-Port Names and Descriptions**

Port Names	Descriptions
DO	Data Output Bus
DOP	Data Output Parity Bus
DI	Data Input Bus
DIP	Data Input Parity Bus
RDADDR	Read Data Address Bus
RDCLK	Read Data Clock
RDEN	Read Port Enable
REGCE	Output Register Clock Enable
RST	Synchronous Set/Reset the output registers/latches
WE	Byte-wide Write Enable
WRADDR	Write Data Address Bus
WRCLK	Write Data Clock
WREN	Write Port Enable

Byte-wide Write Enable

The byte-wide write enable feature of the block RAM gives the capability to write eight bit (one byte) portions of incoming data. There are up to four independent byte-wide write enable inputs to the true dual-port RAM. Each byte-wide write enable is associated with one byte of input data and one parity bit. This feature is useful when using block RAM to interface with a microprocessor. Byte-wide write enable is further described in the Additional Block RAM Primitive Design Considerations section. Figure 7 shows the byte-wide write-enable timing diagram for the block RAM.

When the block RAM is configured for a 18-bit or 9-bit wide data path, any port can restrict writing to specified byte locations within the data word. If configured in READ_FIRST mode, the DO bus shows the previous content of the whole addressed word. In WRITE_FIRST mode, DO shows only the newly written enabled bytes. In NO_CHANGE mode, DO holds the value from the previous clock cycle.

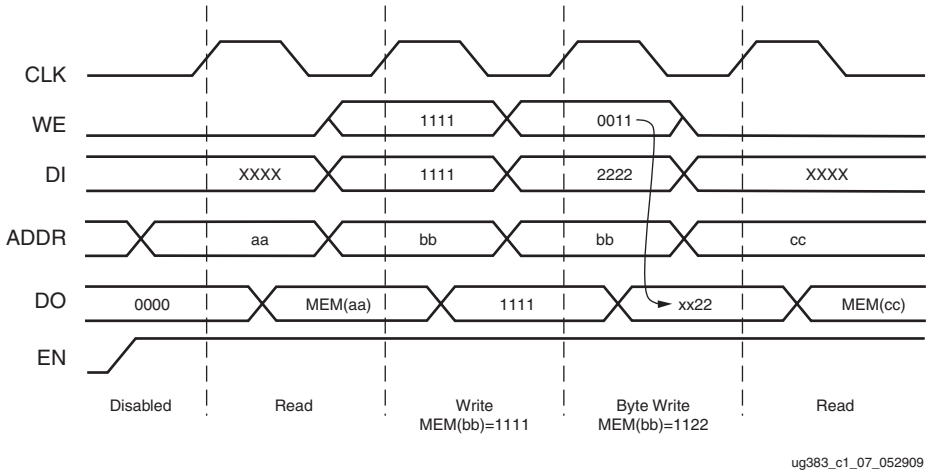


Figure 7: **Byte-wide Write Operation Waveforms (x36 WRITE_FIRST)**

Block RAM Library Primitives

The Spartan-6 FPGA block RAM library primitives, RAMB16BWER and RAMB8BWER, 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. See the Block RAM Attributes section.

Figure 8 illustrates all the I/O ports of the 18 Kb true dual-port block RAM primitive (RAMB16BWER). Figure 9 illustrates the 9 Kb dual-port block RAM primitive (RAMB8BWER). Table 7 lists the 9 Kb and 18 Kb primitives.

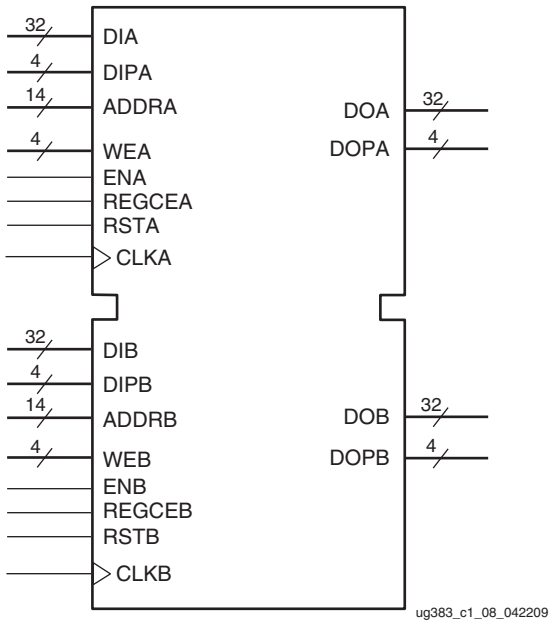


Figure 8: **Block RAM Port Signals (RAMB16BWER)**

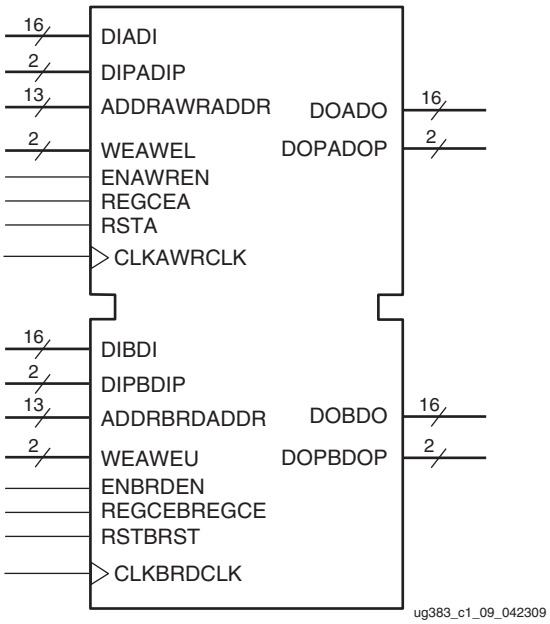


Figure 9: **Block RAM Port Signals (RAMB8BWER)**

Table 7: **Spartan-6 FPGA Block RAM Primitives**

Primitive	Description
RAMB8BWER	Supports data widths of x1, x2, x4, x8, x16, x32 (and x9, x18, x36 with parity bits)
RAMB16BWER	Supports data widths of x1, x2, x4, x8, x16, x32 (and x9, x18, x36 with parity bits)

Notes:

1. All primitives are described in the software Libraries guide as well as the language templates.

Block RAM Port Signals

Each block RAM port operates independently of the other while accessing the same set of 9 Kb or 18 Kb memory cells. The port names for the RAMB16BWER and the RAMB8BWER primitives are not the same. In the following section, the port names for the RAMB16BWER are provided first, then the names for the RAMB8BWER follow in parentheses.

Clock - CLKA, CLKB (CLKAWRCLK, CLKBRDCLK)

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).

For the RAMB8BWER in true-dual port (TDP) mode (RAM_MODE = TDP), CLKAWRCLK and CLKBRDCLK are the clock inputs for the respective ports. In SDP mode (RAM_MODE = SDP), CLKAWRCLK is the write clock input and CLKBRDCLK is the read clock input.

Enable - ENA, ENB (ENAWREN, ENBRDEN)

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).

For the RAMB8BWER in TDP mode (RAM_MODE = TDP), ENAWREN and ENBRDEN are the enable inputs for the respective ports. In SDP mode, ENAWREN is the write enable and ENBRDEN is the read enable.

Byte-wide Write Enable - WEA, WEB (WEAWEL, WEBWEU)

To write the content of the data input bus into the addressed memory location, both EN and WE must be active within a set-up 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 inactive, 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 configurable (default is active High).

For the RAMB8BWER in TDP mode (RAM_MODE = TDP), WEAWEL and WEBWEU are the byte-enable inputs for the respective ports. In SDP mode, WEAWEL is the write enable for the lower 2 bytes and WEBWEU is the write enable for the upper bytes.

Register Enable - REGCEA, REGCEB (REGCEA, REGCEBREGCE)

The register enable pin (REGCE) controls the optional output register. When the RAM is in register mode, REGCE = 1 registers the output into a register at a clock edge. The polarity of REGCE is configurable (default is active High).

For the RAMB8BWER in TDP mode (RAM_MODE = TDP), REGCEA and REGCEBREGCE are the register clock enable inputs for the respective ports. In SDP mode, REGCEA is unused and should be tied to logic 0. REGCEBREGCE is the register enable for the read port.

Set/Reset - RSTA, RSTB (RSTA, RSTBRST)

In latch mode, the RST pin forces the data output latches to contain the value SRVAL. This operation is synchronous or asynchronous depending on the value of the RSTTYPE attribute. See Block RAM Attributes, page 22. When the optional output registers are enabled (DO_REG = 1), the RST signal forces the data output registers contain the SRVAL value. The priority of RST over REGCE is determined using the RST_PRIORITY attribute.The data output registers are asserted to 0 or 1, including the parity bits. The data output latches can also be affected by the RST port based on the EN_RSTRAM attribute. 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. Similar to the read and write operation, the set/reset function is active only when the enable pin of the port is active. The polarity for both signals is configurable (active High by default).

For the RAMB8BWER in TDP mode (RAM_MODE = TDP), RSTA and RSTBRST are the set/reset inputs for the respective ports. In SDP mode, RSTA is unused and should be tied to logic 0. RSTBRST is the set/reset for the read port.

Address Bus - ADDRA, ADDRB (ADDRAWRADDR, ADDRBRDADDR)

The address bus selects the memory cells for read or write. The data bit width of the port determines the required address bus width for a single block RAM, as shown in Table 4. The data width is defined by the DATA_WIDTH_[A|B] attribute.

For the RAMB8BWER in TDP mode (RAM_MODE = TDP), ADDRAWRADDR and ADDRBRDADDR are the enable inputs for the respective ports. In SDP mode, ADDRAWRADDR is the address input for the write port and ADDRBRDADDR is the input for the read port.

Data and address pin mapping is further described in the Additional Block RAM Primitive Design Considerations section.

Data-In Buses - DIA, DIB (DIADI, DIBDI)

Data-in buses provide the new data value to be written into RAM. The regular data-in bus (DI), including the parity bits when available, have a total width equal to the port width. For example, a 36-bit port data width is represented by DI[31:0] as the data bits, and DIP[3:0] as the parity bits, as shown in Table 4. The data bus width is defined by the DATA_WIDTH_[A|B] attribute.

For the RAMB8BWER in TDP mode (RAM_MODE = TDP), DIADI and DIBDI are the data inputs for the respective ports. In SDP mode, DIADI represents the low-order bits and DIBDI represents the high-order bits (when the data width is greater than 16 bits).

Data-In Parity - DIPA, DIPB (DIPADIP, DIPBDIP)

Data-in parity buses provide the new data value to be written to the RAM parity. This area of memory is intended to store parity bits; however, the RAM parity can be used for additional data when needed. No logic is present in the block RAM to calculate or verify the parity bits. In TDP mode DIPADIP and DIPBDIP are the parity data inputs for the respective ports. In SDP mode, DIPADIP is the parity for the lower two bits and DIPBDIP is the parity for the upper two bits.

Data-Out Buses - DOA, DOB (DOADO, DOBDO)

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.

For the RAMB8BWER in TDP mode (RAM_MODE = TDP), DOADO and DOBDO are the outputs for the respective ports. In SDP mode, DOADO represents the low-order bits and DOBDO represents the high-order bits (when the data width is greater than 16).

Data-Out Parity - DOPA, DOPB (DOPADOP, DOPBDOP)

Data-out parity buses reflect the contents of memory cells assigned for parity data. The data appears on the bus with the same behavior as the data-out buses.

For the RAMB8BWER in TDP mode (RAM_MODE = TDP), DOPADOP and DOPBDOP are the outputs for the respective ports. In SDP mode, DOPADOP represents the parity bits for the low-order bytes and DOPBDOP represents the parity bits for the high-order bytes.

GSR

The global set/reset (GSR) signal of a Spartan-6 device is an asynchronous global signal that is active at the end of device configuration. The GSR can also restore the initial Spartan-6 device state at any time. The GSR signal initializes the output latches to the INIT (simple dual port), or to the INIT_A and INIT_B value (true dual port.) See Block RAM Attributes. A GSR signal has no impact on internal memory contents. Because it is a global signal, the GSR has no input pin at the functional level (block RAM primitive).

Unused Inputs

Unused data and/or address inputs should be connected Low.

Block RAM Address Mapping

Each port accesses the same set of 9,216 or 18,432 memory cells using an addressing scheme dependent on whether it is a 9 Kb or 18 Kb RAM. The physical RAM locations addressed for a particular width are determined using the following formula (of interest only when the two ports use different aspect ratios):

```
END = ((ADDR + 1) × Width) -1
START = ADDR × Width
```

Table 8 shows low-order address mapping for each port width.

Table 8: **Port Address Mapping**

Port Width	Parity Locations	Data Locations
1	N.A.	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
2		15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
4		7 6 5 4 3 2 1 0
8 + 1	3 2 1 0	3 2 1 0
16 + 2	1 0	1 0
32 + 4	0	0

Block RAM Attributes

All attribute code examples are discussed in the Block RAM Initialization section. Further information on using these attributes is available in the Additional Block RAM Primitive Design Considerations section.

Data and Address Width - DATA_WIDTH_A, DATA_WIDTH_B

The data bus and address bus for each port is defined by the DATA_WIDTH attribute. The valid combinations of data widths listed are 0 (default),1, 2, 4, 9, 18, and 36. The values of 9, 18, and 36 include the parity bits. The only supported DATA_WIDTH value for a 9 Kb block RAM in simple dual-port mode is 36. For 18 Kb block RAM in either simple or true dual-port mode, or for a 9 Kb block RAM in true dual-port mode, the DATA_WIDTH values can be any combination. The resulting address width, based on data width, is listed in Table 4.

Content Initialization - INIT_xx

INIT_xx attributes define the initial memory contents. By default, block RAM memory is initialized with all zeros during the device configuration sequence. The initialization attributes from INIT_00 through INIT_1F for the 9 Kb block RAM, and from INIT_00 through INIT_3F for the 18 Kb block RAM represent the regular memory contents. Parity memory contents are initialized using the INITP attribute. 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 for determining the bit positions for each INIT_xx attribute.

Given yy = conversion hex-encoded (xx) to decimal, 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 as follows:

- yy = conversion hex-encoded (xx) to decimal “1F” = 31
- from [(31+1) × 256] – 1 = 8191
- to 31 × 256 = 7936

More examples are given in Table 9.

Table 9: **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

Content Initialization - INITP_xx

INITP_xx attributes define the initial contents of the memory cells corresponding to the 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 four initialization attributes are INITP_00 through INITP_03 for the 9 Kb block RAM and INITP_00 through INITP_07 for the 18 Kb block RAM. 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.

Output Latches Initialization - INIT_A, INIT_B

The INIT_A and INIT_B attributes define the output latches or output register values after configuration. These attributes are hex-encoded bit vectors, and the default value is 0. In TDP mode, INIT_A applies to port A and INIT_B applies to port B. In SDP mode, INIT_A represents the low-order output bits and INIT_B represents the high-order output bits.

Output Latches/Registers Synchronous Set/Reset - SRVAL_A, SRVAL_B

The SRVAL_A and SRVAL_B attributes define output latch values when the RST input is asserted. The width of these attributes is the port width. 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. When using parity, the most significant bits correspond to the parity bits.

In TDP mode, SRVAL_A applies to port A and SRVAL_B applies to port B. In SDP mode SRVAL_A applies to the 18 low-order bits (including 2 parity) and SRVAL_B applies to the 18 high-order bits (including 2 parity).

RAM MODE - RAM_MODE

This attribute is only necessary for RAMB8BWER. It determines if the block RAM will function as a true dual-port RAM (two buses for reading and writing) or a simple dual-port RAM (port B dedicated for reading, and port A for writing). Valid values are TDP (default) and SDP.

Reset or CE Priority - RST_PRIORITY_A, RST_PRIORITY_B

When using output registers (DO_REG = 1), this attribute determines the priority of RST or REGCE. When using output latches instead of registers, this attribute determines the priority of RST or EN. Valid values are SR (default) or CE.

Data Latch Reset - EN_RSTRAM_A, EN_RSTRAM_B

The EN_RSTRAM_A and EN_RSTRAM_B attributes determine if the RST port affects the block RAM output latches. Valid values are TRUE or FALSE (default).

Reset Type - RSTTYPE

This attribute determines if the block RAM outputs are reset synchronously or asynchronously. Although the valid values are SYNC (default) and ASYNC, Xilinx recommends using synchronous resets instead of asynchronous resets. For more details, see Reset Behavior.

Optional Output Register On/Off Switch - DO[A|B]_REG

This attribute enables the pipeline register at A/B output of the block RAM. The valid values are 0 (default) or 1.

Write Mode - WRITE MODE_A, WRITE_MODE_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. Additional information on the write modes is in the Write Modes section.

Block RAM Location Constraints

Block RAM instances can have LOC properties attached to them to constrain placement. Block RAM placement locations differ from the convention used for naming CLB locations, allowing LOC properties to transfer easily from array to array. 18 Kb block RAMs use the location identification of RAMB16, to indicate the size of the RAM, not including the parity bits. 9 Kb block RAM use the notation RAMB8 as shown below.

The LOC properties use the following form:

```
LOC = RAMB16_X#Y#;
LOC = RAMB8_X#Y#;
```

The RAMB16_X0Y0 is the bottom-left block 18 Kb RAM location on the device. If an 18 Kb RAM is constrained to RAMB16_X0Y0, a 9 Kb RAM cannot be constrained to RAMB8_X0Y0 or RAMB8_X0Y1 since they share a location.

Block RAM Initialization

Block RAM memory 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. The Spartan-6 FPGA Libraries Guide includes the code to instantiate the 18 Kb and 9 Kb block RAMs.

Additional Block RAM Primitive Design Considerations

Optional Output Registers

Optional output registers can be used at either or both A | B output ports of RAMB16BWER and RAMB8BWER. 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 (RST) pins of ports [A | B] causes the value specified by the attribute SRVAL to be registered at the output. Figure 5 shows an optional output register.

RAMB16BWER and RAMB8BWER Port Mapping Design Rules

The Spartan-6 FPGA block RAM are configurable to various port widths and sizes. Depending on the configuration, some data pins and address pins are not used. Table 4 shows the pins used in various configurations. In addition to the information in Table 4, the following rules are useful to determine port connections for the block RAM:

- When using RAMB16BWER, if the DI[A | B] pins are less than 32-bits wide, concatenate (32 – DI_BIT_WIDTH) logic zeros to the front of DI[A | B]. When using RAMB8BWER, if the DI[A | B] pins are less than 16-bits wide, concatenate (16 – DI_BIT_WIDTH) logic zeros to the front of DI[A | B].
- When using RAMB16BWER, if the DIP[A | B] pins are less than 4-bits wide, concatenate (4 – DIP_BIT_WIDTH) logic zeros to the front of DIP[A | B].
- When using RAMB8BWER, if the DIP[A | B] pins are less than 2-bits wide, concatenate (2 – DIP_BIT_WIDTH) logic zeros to the front of DIP[A | B].
- When using RAMB16BWER, DO[A | B] pins must be 32-bits wide. However, valid data are only found on pins DO_BIT_WIDTH – 1 down to 0.
- When using RAMB16BWER, DOP[A | B] pins must be 4-bits wide. When using RAMB8BWER, DOP[A | B] pins must be 2-bits wide. However, valid data are only found on pins DOP_BIT_WIDTH – 1 down to 0. DOP[A | B] can be left unconnected when not in use.
- When using RAMB16BWER, ADDR[A | B] pins must be 14-bits wide. When using RAMB8BWER, ADDR[A | B] pins must be 13-bits wide. Address width is defined in Table 4.

Reset Behavior

The initial values of the block RAM latches or registers are defined by SR_VAL when RSTTYPE = ASYNC, and INIT when RSTTYPE = SYNC. GSR must be applied after configuration when the reset type is ASYNC and INIT values are used.

Byte-wide Write Enable

The following rules should be considered when using the byte-wide write enable feature:

- In x36 mode, WE[3:0] is connected to the four user WE inputs.
- In x18 mode, WE[0] and WE[2] are connected and driven by the user WE[0], while WE[1], and WE[3] are driven by the user WE[1].
- In x9, x4, x2, x1, WE[3:0] are all connected to a single user WE.

Block RAM Applications

Creating Larger RAM Structures

The Xilinx CORE Generator program offers the designer an easy way to generate wider and deeper memory structures using multiple block RAM instances. This program outputs VHDL or Verilog instantiation templates and simulation models, along with an netlist file for inclusion in a design.

Block RAM RST in Register Mode

A block RAM RST in register mode can be used to control the output register as a true pipeline register independent of the block RAM. As shown in Figure 10, block RAM can be read and written independent of register enable or set/reset. In register mode RST sets DO to the SRVAL and data can be read from the block RAM to DBRAM. Data at DBRAM can be clocked out (DO) on the next cycle. The timing diagrams in Figure 11 and Figure 12 show different cases of the RST operation.

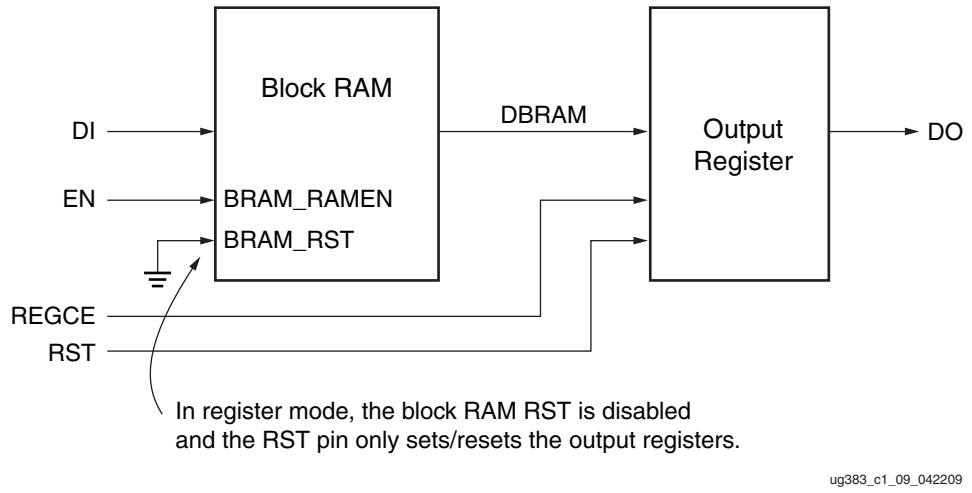


Figure 10: **Block RAM RST in Register Mode**

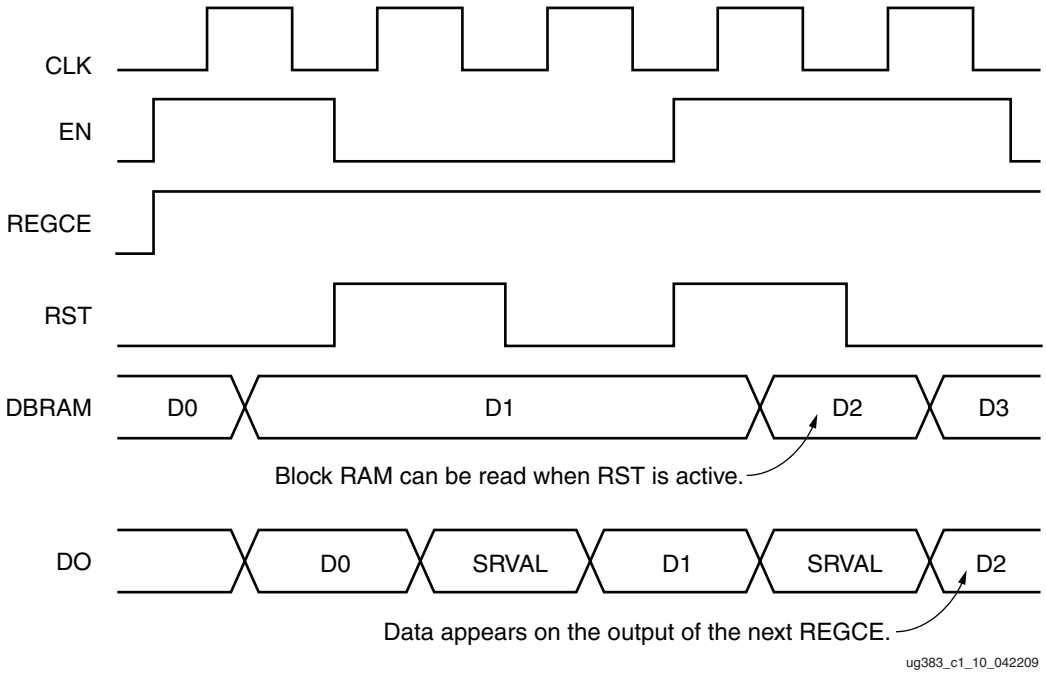


Figure 11: **RST Operation in Register Mode with REGCE High**

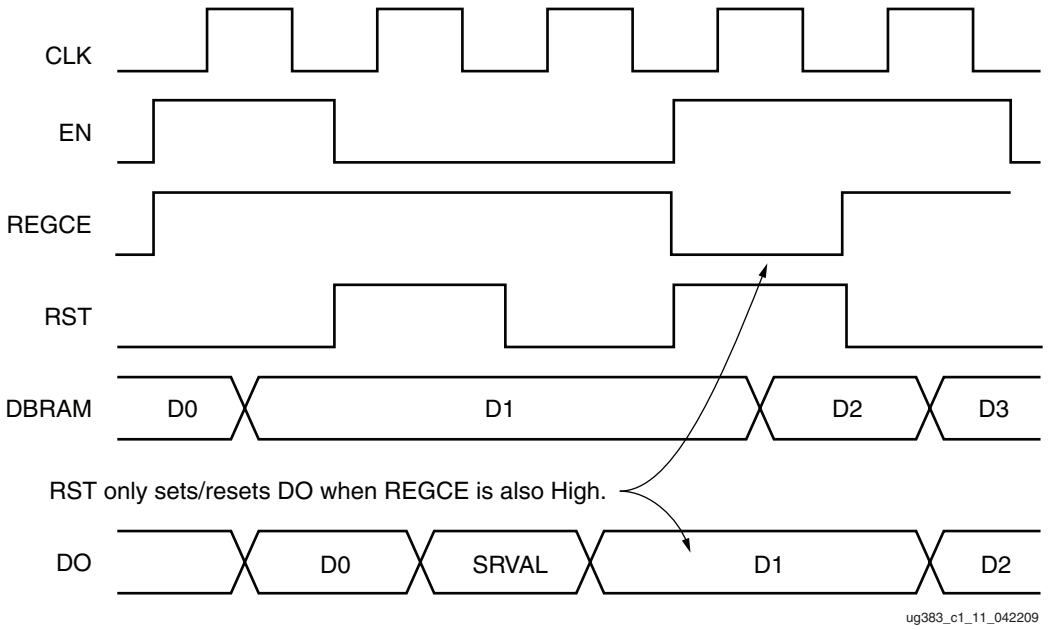


Figure 12: **RST Operation in Register Mode with Variable REGCE**

Block RAM Timing Model

This section describes the timing parameters associated with the block RAM in Spartan-6 devices (illustrated in Figure 13). The switching characteristics section in the *Spartan-6 FPGA Data Sheet* and the Timing Analyzer (TRCE) report from Xilinx software are also available for reference.

Block RAM Timing Parameters

Table 10 shows the Spartan-6 FPGA block RAM timing parameters.

Table 10: **Block RAM Timing Parameters**

Parameter	Function	Control Signal	Description
Setup and Hold Relative to Clock (CLK)			
T_{RxCK_x} = Setup time (before clock edge) and T_{RCKx_x} = Hold time (after clock edge)			
T_{RCCK_ADDR}	Address inputs	ADDR	Time before the clock that address signals must be stable at the ADDR inputs of the block RAM.(1)
T_{RCKC_ADDR}			Time after the clock that address signals must be stable at the ADDR inputs of the block RAM.(1)
T_{RDCK_DI}	Data inputs	DI	Time before the clock that data must be stable at the DI inputs of the block RAM.
T_{RCKD_DI}			Time after the clock that data must be stable at the DI inputs of the block RAM.

Table 10: **Block RAM Timing Parameters *(Cont'd)***

Parameter	Function	Control Signal	Description
T_{RCCK_EN}	Enable	EN	Time before the clock that the enable signal must be stable at the EN input of the block RAM.
T_{RCKC_EN}			Time after the clock that the enable signal must be stable at the EN input of the block RAM.
T_{RCCK_RST}	Synchronous Set/Reset	RST	Time before the clock that the synchronous set/reset signal must be stable at the RST input of the block RAM.
T_{RCKC_RST}			Time after the clock that the synchronous set/reset signal must be stable at the RST input of the block RAM.
T_{RCCK_WE}	Write Enable	WE	Time before the clock that the write enable signal must be stable at the WE input of the block RAM.
T_{RCKC_WE}			Time after the clock that the write enable signal must be stable at the WE input of the block RAM.
T_{RCCK_REGCE}	Optional Output Register Enable	REGCE	Time before the CLK that the register enable signal must be stable at the REGCE input of the block RAM.
T_{RCKC_REGCE}			Time after the clock that the register enable signal must be stable at the REGCE input of the block RAM.
Clock to Out Delays			
T_{RCKO_DO} (latch mode)	Clock to Output	CLK to DO	Time after the clock that the output data is stable at the DO outputs of the block RAM (without output register).
$T_{RCKO_DO_REG}$ (register mode)	Clock to Output	CLK to DO	Time after the clock that the output data is stable at the DO outputs of the block RAM (with output register).

Notes:

1. While EN is active, ADDR inputs must be stable during the entire setup/hold time window, even if WE is inactive. Violating this requirement can result in block RAM data corruption. If ADDR timing could violate the specified requirements, EN must be inactive (disabled).

Block RAM Timing Characteristics

The timing diagram in Figure 13 describes a single-port block RAM in write-first mode without the optional output register. The timing for read-first and no-change modes are similar. For timing using the optional output register, an additional clock latency appears at the DO pin. These waveforms correspond to latch mode when the optional output pipeline register is not used.

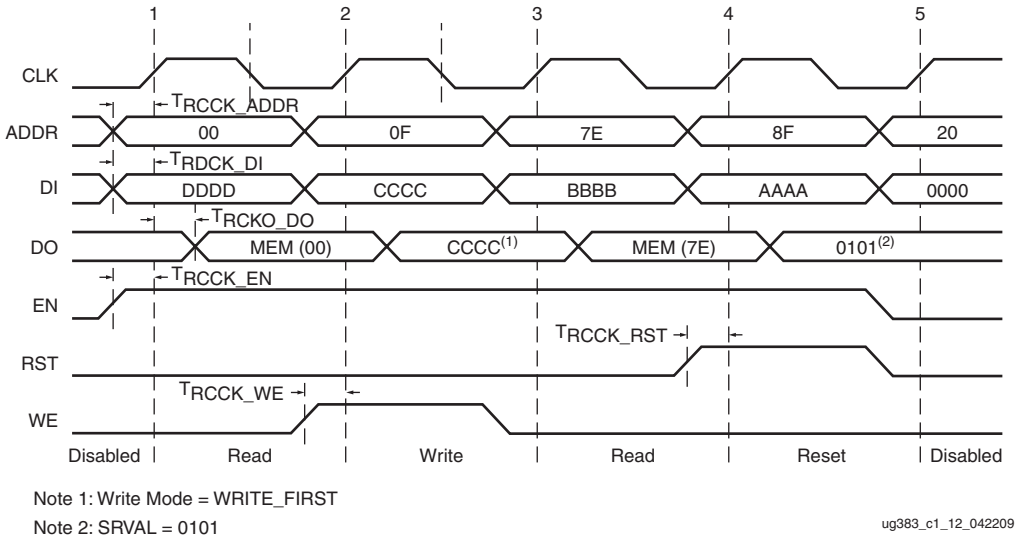


Figure 13: **Block RAM Timing Diagram**

At time 0, the block RAM is disabled; EN (enable) is Low.

Clock Event 1

Read Operation

During a read operation, the contents of the memory at the address on the ADDR inputs remain unchanged.

- T_{RCCK_ADDR} before clock event 1, address `00` becomes valid at the ADDR inputs of the block RAM.
- At time T_{RCCK_EN} before clock event 1, enable is asserted High at the EN input of the block RAM, enabling the memory for the READ operation that follows.
- At time T_{RCKO_DO} after clock event 1, the contents of the memory at address `00` become stable at the DO pins of the block RAM.
- Whenever EN is asserted, all address changes must meet the specified setup and hold window. Asynchronous address changes can affect the memory content and block RAM functionality in an unpredictable way.

Clock Event 2

Write Operation

During a write operation, the content of the memory at the location specified by the address on the ADDR inputs is replaced by the value on the DI pins and is immediately reflected on the output latches (in WRITE_FIRST mode); when Write Enable (WE) is High.

- At time T_{RCCK_ADDR} before clock event 2, address `0F` becomes valid at the ADDR inputs of the block RAM.
- At time T_{RDCK_DI} before clock event 2, data `CCCC` becomes valid at the DI inputs of the block RAM.
- At time T_{RCCK_WE} before clock event 2, write enable becomes valid at the WE following the block RAM.
- At time T_{RCKO_DO} after clock event 2, data `CCCC` becomes valid at the DO outputs of the block RAM.

Clock Event 4

RST (Synchronous Set/Reset) Operation

During an RST operation, initialization parameter value SRVAL is loaded into the output latches of the block RAM. The RST operation does NOT change the contents of the memory and is independent of the ADDR and DI inputs.

- At time T_{RCCK_RST} before clock event 4, the synchronous set/reset signal becomes valid (High) at the RST input of the block RAM.
- At time T_{RCKO_DO} after clock event 4, the SRVAL `0101` becomes valid at the DO outputs of the block RAM.

Clock Event 5

Disable Operation

Deasserting the enable signal EN disables any write, read, or RST operation. The disable operation does NOT change the contents of the memory or the values of the output latches.

- At time T_{RCCK_EN} before clock event 5, the enable signal becomes invalid (Low) at the EN input of the block RAM.
- After clock event 5, the data on the DO outputs of the block RAM is unchanged.

Block RAM Timing Model

Figure 14 illustrates the delay paths associated with the implementation of block RAM. This example takes the simplest paths on and off chip (these paths can vary greatly depending on the design). This timing model demonstrates how and where the block RAM timing parameters are used.

- **NET** = Varying interconnect delays
- $\mathbf{T_{IOPI}}$ = Pad to I-output of IOB delay
- $\mathbf{T_{IOOP}}$ = O-input of IOB to pad delay
- $\mathbf{T_{BUFG}}$ = BUFG delay

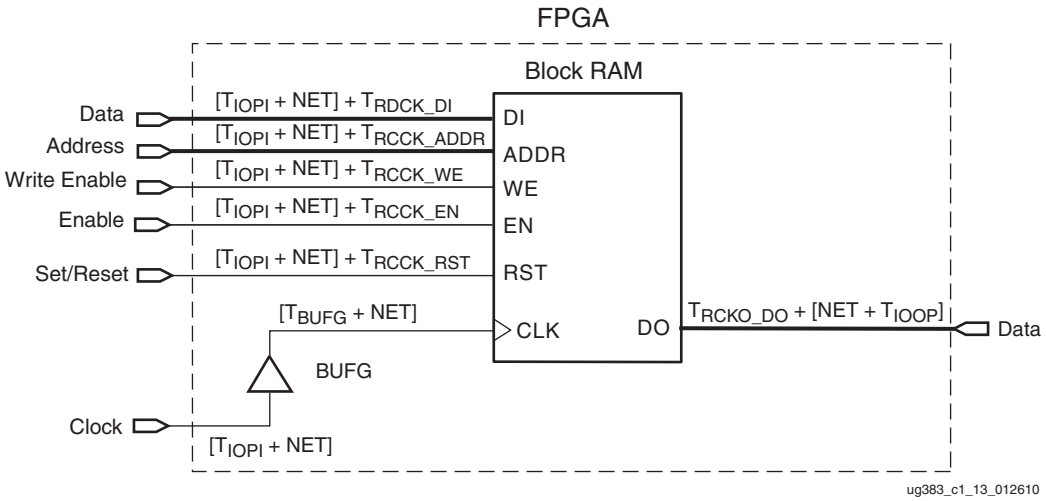


Figure 14: **Block RAM Timing Model**