IP CORE MANUAL

AXI4–Stream BRAM to Wave 256 IP

px_axis_bram2wave_256

PENTEK

Pentek, Inc.
One Park Way
Upper Saddle River, NJ 07458
(201) 818–5900
http://www.pentek.com/

Manual Part Number: 807.48435 Rev: 1.0 – November 30, 2018

Manual Revision History

Date	Version	Comments
11/30/18	1.0	Initial Release

Legal Notices

Copyright

Trademarks

Printed in the United States of America.

Table of Contents

Page

Table of Contents

Page

Chapter 4: Register Space

Table of Contents

Page

Chapter 5: Designing with the Core

Chapter 6: Design Flow Steps

Table of Contents

Page

This page is intentionally blank

IP Facts

Description

Pentek's Navigator™ AXI4–Stream BRAM to Wave 256 Core provides a means of generating an AXI4–Stream waveform from wavetable data stored in an external BRAM. Trigger length and delay are user–configurable via an AXI4–Lite CSR interface, as are output data format and trigger mode.

The AXI4–Stream BRAM to Wave Core requires an External Dual–Port BRAM for the waveform table. The waveform table is written to the BRAM by the user design utilizing a separate, external BRAM interface.

This core complies with the ARM® AMBA® AXI4 Specification. This user manual defines the hardware interface, software interface, and parameterization options for the AXI4–Stream BRAM to Wave 256 Core.

Features

- User–configurable trigger length and delay via the AXI4–Lite CSR interface
- User–selectable data mode (Time Packed Real or Channel Packed I/Q Data) via the AXI4–Lite CSR interface
- User–selectable status, mode select for trigger via the AXI4–Lite CSR interface
- Interrupts are generated for acquisition start/end, arm status, bad trigger and trigger start

Table 1–1: IP Facts Table

Core Specifics	
Supported Design Family[a]	Zynq® Ultrascale+ RFSoC
Supported User Interfaces	AXI4–Lite and AXI4–Stream
Resources	See Table 2–1
Provided with the Core	
Design Files	VHDL
Example Design	Not Provided
Test Bench	VHDL
Constraints File	Not Provided[b]
Simulation Model	VHDL
Supported S/W Driver	HAL Software Support
Tested Design Flows	
Design Entry	Vivado® Design Suite 2018.3 or later
Simulation	Vivado VSim
Synthesis	Vivado Synthesis
Support	
Provided by Pentek fpgasupport@pentek.com	

a.For a complete list of supported devices, see the *Vivado Design Suite Release Notes*.
b.Clock constraints can be applied at the top–level module of the user design.

This page is intentionally blank

Chapter 1: Overview

1.1 Functional Description

The AXI4−Stream BRAM to Wave 256 Core generates an AXI4−Stream waveform from a wave table stored in an external Dual−Port BRAM. The trigger length as well as the delay from the trigger can be controlled by the user via an AXI4−Lite CSR interface. Trigger mode and output data format (i.e. data mode) are also user configurable via this interface.

The waveform table is generated outside of the core, and is written to the external Dual−Port BRAM through an external BRAM controller in the user design. The AXI4−Stream BRAM to Wave 256 Core only has read access to this wavetable memory.

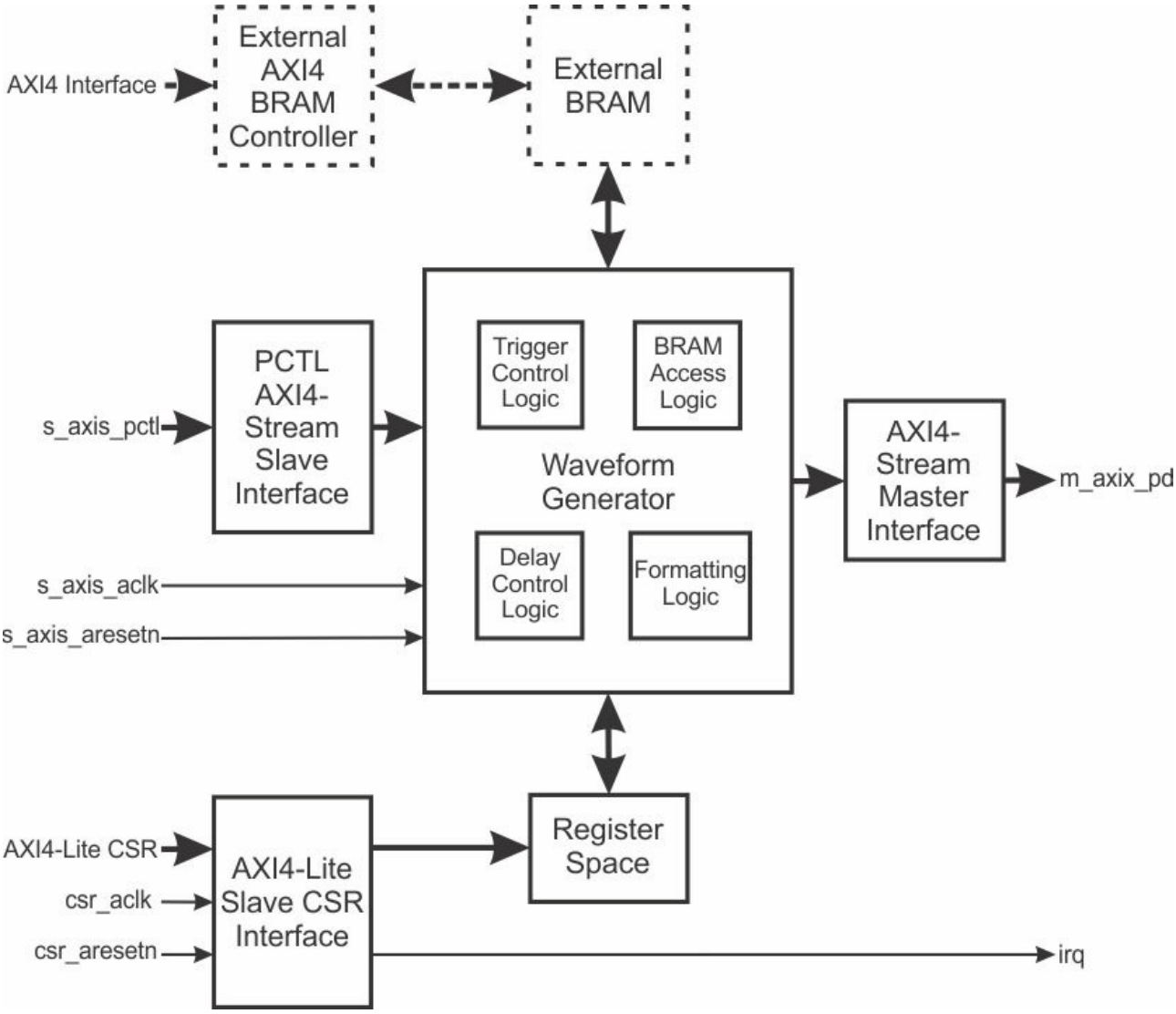
NOTE: The waveform table length **must** be a multiple of 16.

NOTE: The trigger length must be a multiple of 16 when the Data Mode (see Section 4.1) is set to "Time Packed Real Data" mode, or a multiple of 8 when set to either "Packed I/Q Data or 2−Channel Real Data" mode.

Figure 1−1 is a top−level block diagram of the AXI4−Stream BRAM to Wave 256 Core. The modules within the block diagram are explained in other sections of this manual.

1.1 Functional Description (continued)

Figure 1–1: AXI4–Stream BRAM to Wave 256 Core Block Diagram



- **External AXI4 BRAM Controller:** This module is external to the core and provides write access to the External BRAM. Wave tables are written to the BRAM by the user design via this interface.

- **External AXI4 BRAM:** This is the Xilinx BRAM core which holds the waveform table. The External AXI4 BRAM controller module has write access to this memory, while the AXI4–Stream BRAM to Wave Core only has read access. For details on generating the core, refer to Section 5.2.

1.1 Functional Description (continued)

- **PCTL AXI4–Stream Slave Interface:** This module implements an AXI4–Stream Slave PCTL–style interface, providing the input triggers for the core. For additional details about the AXI4–Stream Slave Interface, refer to Section 3.2.1.

- **Waveform Generator:** This module generates the waveform based on the parameters set by the user and the data in the waveform table. The waveform data is then formatted per the user–accessible register settings and then passed to the AXI4–Stream Master Interface.

- **AXI4–Stream Master Interface:** This module implements an AXI4–Stream Master interface for the output data stream for the core. For additional details about the AXI4–Stream Slave Interface, refer to Section 3.2.2.

- **AXI4–Lite Slave CSR Interface:** This module implements a 32–bit AXI4–Lite Slave interface to access the Register Space. For additional details about the AXI4–Lite Interface, refer to the Section 3.1.1.

- **Register Space:** This module contains the control registers, status registers and interrupts for the core. The register space is accessed through the AXI4–Lite CSR interface. See Chapter 4 for the register memory map and bit mapping details.

1.2 Applications

This core is useful for generating AXI4–Stream waveforms from a wave table.

1.3 System Requirements

For a list of system requirements, see the Vivado Design Suite Release Notes.

1.4 Licensing and Ordering Information

This core is included with all Pentek Navigator FPGA Design Kits for Pentek Jade series board products. Contact Pentek for Licensing and Ordering Information (www.pentek.com).

1.5 Contacting Technical Support

Technical Support for Pentek's Navigator FPGA Design Kits is available via e–mail (fpgasupport@pentek.com) or by phone (201–818–5900 ext. 238, 9 am to 5 pm EST).

1.6 Documentation

This user manual is the main document for this IP core. The following documents provide supplemental material:

1) *Vivado Design Suite User Guide: Designing with IP*

2) *Vivado Design Suite User Guide: Programming and Debugging*

3) *Vivado Design Suite: FIFO Generator LogiCore IP Product Guide*

4) *ARM AMBA AXI4 Protocol Version 2.0 Specification*
http://www.arm.com/products/system–ip/amba–specifications.php

Chapter 2: General Product Specifications

2.1 Standards

The AXI4–Stream BRAM to Wave 256 Core has bus interfaces that comply with the *AMBA AXI4–Lite Protocol Specification* and the *AMBA AXI4–Stream Protocol Specification*.

2.2 Performance

The performance of the AXI4–Stream BRAM to Wave 256 Core is limited primarily by the FPGA logic speed. The values presented in this section should be used as an estimation guideline. Actual performance can vary.

2.2.1 Maximum Frequencies

The AXI4–Stream BRAM to Wave 256 Core has two incoming clock signals, the AXI4–Stream clock (**`s_axis_aclk`**) and AXI4–Lite Interface CSR clock (**`s_axi_csr_aclk`**). The AXI4–Lite Interface CSR clock has a maximum frequency of 250 MHz, and the AXI4–Stream clock has a maximum frequency of 500 MHz on a Zynq Ultrascale+ RFSOC –2 speed grade FPGA. Note that 250 MHz is typically the PCI Express (PCIe) AXI bus clock frequency.

2.3 Resource Utilization

The resource utilization of the AXI4–Stream BRAM to Wave 256 Core is shown in Table 2–1. Resources have been estimated for a core with an address width of 10 (default). The target device is a Zynq Ultrascale+ RFSOC XCZU27dr –2 speed grade device. These values were generated using the Vivado Design Suite.

Table 2–1: Resource Usage and Availability

Resource	# Used
LUTs	435
Flip–Flops	1,561

NOTE: Actual utilization may vary based on the user design in which the AXI4–Stream BRAM to Wave 256 Core is incorporated.

NOTE: Resources required for the External BRAM and BRAM Controller ***are not*** included in the estimate.

2.4 Limitations and Unsupported Features

- The waveform table length **must** be a multiple of 16.
- The trigger length must be a multiple of 16 when the Data Mode (see Section 4.1) is set to "Time Packed Real Data" mode, or a multiple of 8 when set to either "Packed I/Q Data or 2–Channel Real Data" mode.

2.5 Generic Parameters

There is one generic parameter for the AXI4–Stream BRAM to Wave 256 Core, and it is described in Table 2–2. This parameter can be set as required by the user application while customizing the core.

Table 2-2: Generic Parameters

Port/Signal Name	Type	Description
addr_width	Integer	**Address Width:** This parameter defines the width of the address bus to the external wave table BRAM. The acceptable range is 4 to 31, the default is 10.

Chapter 3: Port Descriptions

This chapter provides port descriptions for the following interface types:

- AXI4–Lite Core Interfaces
- AXI4–Stream Core Interfaces
- I/O Signals

3.1 AXI4–Lite Core Interfaces

The AXI4–Stream BRAM to Wave 256 Core uses the Control/Status Register (CSR) interface to access the Register Space from the user design.

3.1.1 Control/Status Register (CSR) Interface

The CSR interface is an AXI4–Lite Slave Interface that can be used to access the control registers in the AXI4–Stream BRAM to Wave 256 Core. Table 3–1 defines the ports in the CSR Interface. See Chapter 4 for a Register memory map and bit definitions. See the *AMBA AXI4–Lite Specification* for more details on the AXI4–Lite interface.

Table 3-1: Control/Status Register (CSR) Interface Port Descriptions

Port	Direction	Width	Description
s_axi_csr_aclk	Input	1	**Clock**
s_axi_csr_aresetn	Input	1	**Reset:** Active low. This value will reset all control registers to their initial states.
s_axi_csr_awaddr	Input	12	**Write Address:** Address used for write operations. It must be valid when **s_axi_csr_awvalid** is asserted and must be held until **s_axi_csr_awready** is asserted by the AXI4–Stream BRAM to Wave 256 Core.
s_axi_csr_awprot	Input	3	**Protection:** The AXI4–Stream BRAM to Wave 256 Core ignores these bits.
s_axi_csr_awvalid	Input	1	**Write Address Valid:** This input must be asserted to indicate that a valid write address is available on **s_axi_csr_awaddr**. The AXI4–Stream BRAM to Wave 256 Core asserts **s_axi_csr_awready** when it is ready to accept the address. The **s_axi_csr_awvalid** must remain asserted until the rising clock edge after the assertion of **s_axi_csr_awready**.

Table 3-1: Control/Status Register (CSR) Interface Port Descriptions (Continued)			
Port	**Direction**	**Width**	**Description**
s_axi_csr_awready	Output	1	**Write Address Ready:** This output is asserted by the AXI4–Stream BRAM to Wave 256 Core when it is ready to accept the write address.The address is latched when **s_axi_csr_awvalid** and **s_axi_csr_awready** are high on the same cycle.
s_axi_csr_wdata	Input	32	**Write Data:** This data will be written to the address specified by **s_axi_csr_awaddr** when **s_axi_csr_wvalid** and **s_axi_csr _wready** are both asserted. The value must be valid when **s_axi_csr_wvalid** is asserted and held until **s_axi_csr_wready** is also asserted.
s_axi_csr_wstrb	Input	4	**Write Strobes:** This signal, when asserted, indicates the number of bytes of valid data on the **s_axi_csr_wdata** signal. Each of these bits, when asserted, indicate that the corresponding byte of **s_axi_csr_wdata** contains valid data. Bit 0 corresponds to the least significant byte, and bit 3 to the most significant.
s_axi_csr_wvalid	Input	1	**Write Valid:** This signal must be asserted to indicate that the write data is valid for a write operation. The value on **s_axi_csr _wdata** is written into the register at address **s_axi_csr_awaddr** when **s_axi_csr_wready** and **s_axi_csr_wvalid** are high on the same cycle.
s_axi_csr_wready	Output	1	**Write Ready:** This signal is asserted by the AXI4–Stream BRAM to Wave 256 Core when it is ready to accept data. The value on **s_axi_csr_wdata** is written into the register at address **s_axi_csr_awaddr** when **s_axi_csr_wready** and **s_axi_csr_wvalid** are high on the same cycle, assuming that the address has already or simultaneously been submitted.
s_axi_csr_bresp	Output	2	**Write Response:** The AXI4–Stream BRAM to Wave 256 Core indicates success or failure of a write transaction through this signal, which is valid when **s_axi_csr_bvalid** is asserted; 00 = Success of normal access 01 = Success of exclusive access 10 = Slave Error 11 = Decode Error Note: For more details about this signal refer to the *AMBA AXI Specification*.
s_axi_csr_bready	Input	1	**Write Response Ready:** This signal must be asserted by the user logic when it is ready to accept the Write Response.
s_axi_csr_bvalid	Output	1	**Write Response Valid:** This signal is asserted by the AXI4–Stream BRAM to Wave 256 Core when the write operation is complete and the Write Response is valid. It is held until **s_axi_csr_bready** is asserted by the user logic.

Table 3-1: Control/Status Register (CSR) Interface Port Descriptions (Continued)			
Port	**Direction**	**Width**	**Description**
s_axi_csr_araddr	Input	12	**Read Address:** Address used for read operations. It must be valid when **s_axi_csr_arvalid** is asserted and must be held until **s_axi_csr_arready** is asserted by the AXI4–Stream BRAM to Wave 256 Core.
s_axi_csr_arprot	Input	3	**Protection:** These bits are ignored by the AXI4–Stream BRAM to Wave 256 Core.
s_axi_csr_arvalid	Input	1	**Read Address Valid:** This input must be asserted to indicate that a valid read address is available on **s_axi_csr_araddr**. The core asserts **s_axi_csr_arready** when it ready to accept the Read Address. This input must remain asserted until the rising clock edge after the assertion of **s_axi_csr_arready**.
s_axi_csr_arready	Output	1	**Read Address Ready:** This output is asserted by the AXI4–Stream BRAM to Wave 256 Core when it is ready to accept the read address. The address is latched when **s_axi_csr_arvalid** and **s_axi_csr_arready** are high on the same cycle.
s_axi_csr_rdata	Output	32	**Read Data:** This value is the data read from the address specified by the **s_axi_csr_araddr** when **s_axi_csr_arvalid** and **s_axi_csr_arready** are high on the same cycle.
s_axi_csr_rresp	Output	2	**Read Response:** The AXI4–Stream BRAM to Wave 256 Core indicates success or failure of a read transaction through this signal, which is valid when **s_axi_csr_rvalid** is asserted; 00 = Success of normal access 01 = Success of exclusive access 10 = Slave Error 11 = Decode Error Note: For more details about this signal refer to the *AMBA AXI Specification*.
s_axi_csr_rvalid	Output	1	**Read Data Valid:** This signal is asserted by the AXI4–Stream BRAM to Wave 256 Core when the read is complete and the read data is available on **s_axi_csr_rdata**. It is held until **s_axi_csr_rready** is asserted by the user logic.
s_axi_csr_rready	Input	1	**Read Data Ready:** This signal is asserted by the user logic when it is ready to accept the Read Data.
irq	Output	1	**Interrupt:** This is an active high, edge–type interrupt output.

3.2 AXI4–Stream Core Interfaces

The AXI4–Stream BRAM to Wave 256 Core has the following AXI4–Stream Interfaces, used to receive and transfer data streams:

3.2.1 PCTL AXI4–Stream Slave Interface

This interface provides the gate/trigger input for the AXI4–Stream BRAM to Wave 256 Core. Table 3–2 defines the ports in the PCTL AXI4–Stream Slave Interface. See the *AMBA AXI4–Stream Protocol Specification* for more details on the operation of the AXI4–Stream Interface.

Table 3-2: PCTL AXI4–Stream Slave Interface Port Descriptions

Port/Signal Name	Direction	Width	Description
s_axis_aclk	Input	1	**AXI4-Stream Clock**
s_axis_aresetn	Input	1	**Reset:** Active low
s_axis_pctl_tvalid	Input	1	**Input Data Valid:** This signal is asserted by the user logic when data is valid on **`s_axis_pctl_tdata`** bus.
s_axis_pctl_tdata	Input	32	**Input Data:** Input control from the Sync Bus is contained in this bus. The relevant bits are as follows: Bit 0 = Gate Bit 8 = Sync Bit 16 = PPS

3.2.2 AXI4–Stream Master Interface

This interface is the output data bus for the AXI4–Stream BRAM to Wave Core. Table 3–3 defines the ports in the AXI4–Stream Master Interface. See the *AMBA AXI4–Stream Protocol Specification* for more details on the operation of the AXI4–Stream Interface

Table 3-3: AXI4–Stream Master Interface Port Descriptions

Port/Signal Name	Direction	Width	Description
m_axis_pd_tvalid	**Input**	1	**Output Data Valid:** This signal is asserted by the AXI4-Stream BRAM to Wave Core when data is valid on **`m_axis_pd_tdata bus`**.
m_axis_pd_tdata	**Input**	256	**Output Data:** This is the output data bus containing the generated waveform. When "**`data_mode`**" is set to "Time Packed Real Data, the data is formatted in sample pairs as follows: (I[t1], I[t0]). When "**`data_mode`**" is set to "Channel Packed I/Q or 2-Channel Real" mode, the data is formatted as (Q0, I0) or (Ib, Ia).

3.3 I/O Signals

The I/O port/signal descriptions of the top level module of the AXI4–Stream BRAM to Wave 256 Core are discussed in Table 3–4.

Table 3-4: I/O Signals

Port/ Signal Name	Type	Direction	Description
bram_rdport_clk	std_logic	Output	**BRAM Clock:** This is the clock for the read port of the external BRAM.
bram_rdport_rst	std_logic	Output	**BRAM Reset:** This active HIGH reset output for the external BRAM is associated with **`bram_rdport_clk`**.
bram_rdport_en	std_logic	Output	**BRAM Enable:** This is the active HIGH enable for the read port of the external BRAM.
bram_rdport_addr	std_logic _vector	Output	**BRAM Address:** This is the address bus for the read port of the external BRAM. The width of this bus is set by the **`addr_width`** parameter (see **Section 2.5**).
bram_rdport _rddata	std_logic	Output	**BRAM Read Data:** This is the data bus from the read port of the external BRAM.

This page is intentionally blank

Chapter 4: Register Space

This chapter provides the memory map and register descriptions for the Register Space of the AXI4–Stream BRAM to Wave 256 Core. The memory map is provided in Table 4–1.

Table 4–1: Register Space Memory Map

Register Name	Address (Base Address +)	Access	Description
Mode Control	0x00	R/W	This register contains settings for trigger, gate, and data mode.
Trigger Clear	0x04	R/W	This register provides a reset for the trigger.
BRAM Start Address	0x08	R/W	This register provides the BRAM Start Address. **Note:** The address range set by the BRAM Start Address register and the BRAM End Address register must be a multiple of 16.
BRAM End Address	0x0C	R/W	This register provides the BRAM End Address. **See note above.**
Trigger Delay	0x10	R/W	This register sets the delay from trigger to start of waveform.
Trigger Length	0x14	R/W	This register sets the trigger length for the core. **Note:** Trigger length must be a multiple of 16 when Data Mode is set to "Time Packed Real Data" mode, otherwise it must be a multiple of 8.
Status	0x18	RO	This register provides the "arm" status as well as the trigger and data mode settings.
Reserved	0x1C	N/A	Reserved
Reserved	0x20	N/A	Reserved
Interrupt Enable	0x24	R/W	This register provides the interrupt enable settings.
Interrupt Status	0x28	RO	This register provides the interrupt status.
Interrupt Flag	0x2C	R/CLR	This register provides the status of the interrupt flags.

4.1 Mode Control Register

This register provides the mode control settings for the core. This register is illustrated in Figure 4–1 and described in Table 4–2.

Figure 4–1: Mode Control Register

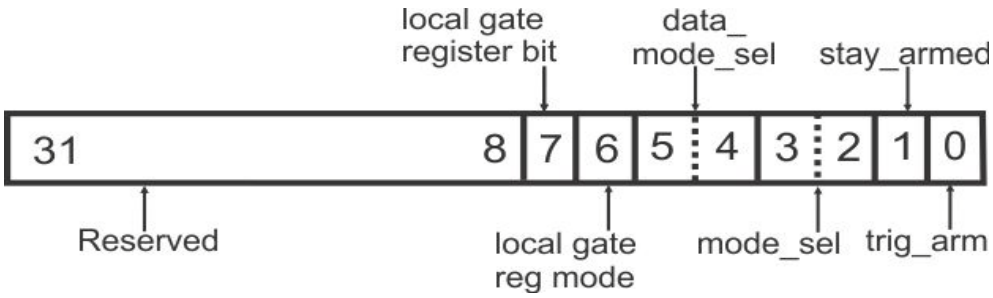


Table 4–2: Mode Control Register (Base Address + 0x00)

Bits	Field Name	Default Value	Access Type	Description
31:8	Reserved	N/A	N/A	**Reserved**
7	local gate register bit	0	R/W	**Local Gate:** When **Local Gate Reg Mode** bit (bit 6) is HIGH, setting this bit HIGH provides a single–pulse gate signal for the trigger.
6	local gate reg mode	0	R/W	**Local Gate Reg Mode:** When HIGH, the **Local Gate Register** bit (bit 7) becomes the gate trigger.
5:4	data_mode_sel	00	R/W	**Data Mode Select:** Selects the format of the output data stream as follows: 00 = Time Packed Real Data 01 = Packed I/Q Data or 2–Channel Real Data 10 = Reserved 11 = Reserved
3:2	mode_sel	00	R/W	**Mode Select:** Selects the trigger mode as follows: 00 = Gate Mode 01 = Trig Mode 10 = Trig Hold Mode 11 = Reserved
1	stay_armed	0	R/W	**Stay Armed:** When HIGH, the trigger remains armed until it is either cleared or manually disarmed.
0	trig_arm	0	R/W	**Trigger Arm:** Set this bit HIGH to arm the trigger.

4.2 Trigger Clear Register

This register provides a clear for the trigger. This register is illustrated in Figure 4–2 and described in Table 4–3.

Figure 4–2: Trigger Clear Register

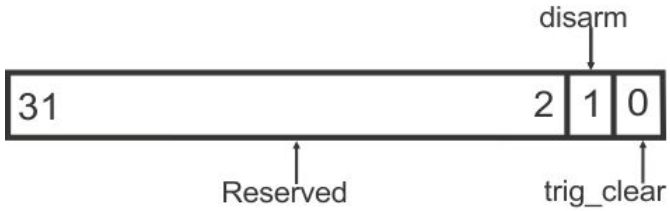


Table 4–3: Trigger Clear Register (Base Address + 0x04)

Bits	Field Name	Default Value	Access Type	Description
31:2	Reserved	N/A	N/A	**Reserved**
1	disarm	0	R/W	**Disarm:** When HIGH, the trigger is disarmed AFTER a trigger is reached.
0	trig_clear	0	R/W	**Trigger Clear:** Set HIGH to clear the trigger immediately.

4.3 BRAM Start Address Register

This register provides the BRAM start address for the wavetable. This register is illustrated in Figure 4–3 and described in Table 4–4.

Figure 4–3: BRAM Start Address Register

Table 4–4: BRAM Start Address Register (Base Address + 0x08)

Bits	Field Name	Default Value	Access Type	Description
31:0	BRAM Start Address	0x00000000	R/W	**BRAM Start Address:** This register provides the starting address for the wavetable in the BRAM. **Note:** The address range set by the BRAM Start Address register and the BRAM End Address register must be a multiple of 16.

4.4 BRAM End Address Register

This register provides the BRAM end address for the wavetable. This register is illustrated in Figure 4–4 and described in Table 4–5.

Figure 4–4: BRAM End Address Register

Table 4–5: BRAM End Address Register (Base Address + 0x0C)

Bits	Field Name	Default Value	Access Type	Description
31:0	BRAM End Address	0x00000000	R/W	**BRAM End Address:** This register provides the ending address for the wavetable in the BRAM. **Note:** The address range set by the BRAM Start Address register and the BRAM End Address register must be a multiple of 16.

4.5 Trigger Delay Register

This register provides the Trigger Delay setting for the core. This register is illustrated in Figure 4–5 and described in Table 4–6.

Figure 4–5: Trigger Delay Register

Table 4–6: Trigger Delay Register (Base Address + 0x10)

Bits	Field Name	Default Value	Access Type	Description
31:0	Trigger Delay	0x00000000	R/W	**Trigger Delay:** This register provides the delay value (in **`s_axis_aclk`** periods) for the trigger hold–off.

4.6 Trigger Length Register

This register provides the Trigger Length setting for the core. This register is illustrated in Figure 4–6 and described in Table 4–7.

Figure 4–6: Trigger Length Register

Table 4–7: Trigger Length Register (Base Address + 0x14)

Bits	Field Name	Default Value	Access Type	Description
31:0	Trigger Length	0x0000 0000	R/W	**Trigger Length:** This register provides the trigger length for the waveform. **Note:** The trigger length must be a multiple of 16 when Data Mode (see Section 4.1) is set to "Time Packed Real Data", otherwise it must be a multiple of 8.

4.7 Status Register

This register provides the status for the core. This register is illustrated in Figure 4–7 and described in Table 4–8.

Figure 4–7: Status Register

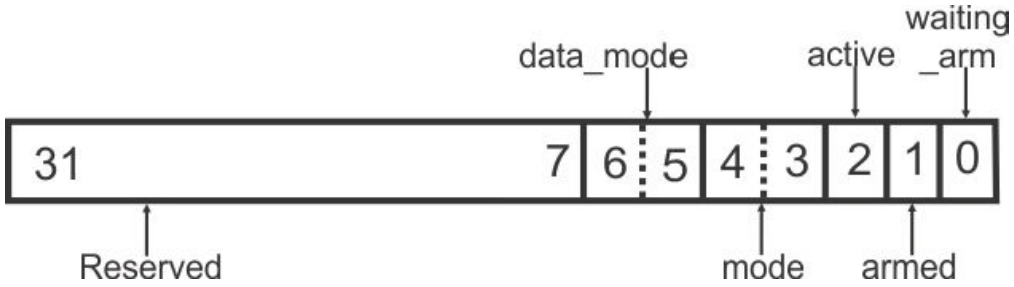


Table 4–8: Status Register (Base Address + 0x18)

Bits	Field Name	Default Value	Access Type	Description
31:7	Reserved	N/A	N/A	**Reserved**
6:5	data_mode	00	RO	**Data Mode:** These register bits provide the current setting of the data mode as follows: 00 = Time Packed Real Data Mode 01 = Packed I/Q Data or 2–Channel Real Data Mode 10 = Reserved 11 = Reserved
4:3	mode	00	RO	**Trigger Mode:** These register bits provide the current setting of the trigger mode as follows: 00 = GATE_MODE 01 = TRIG_MODE 10 = TRIG_HOLD_MODE 11 = Reserved
2	active	0	RO	**Active:** When HIGH, this register bit indicates that a trigger was found and the waveform is being generated.
1	armed	0	RO	**Armed:** When HIGH, this register bit indicates that the trigger is armed.
0	waiting_arm	0	RO	**Waiting for Arm:** When HIGH, this register bit indicates that the trigger is waiting to be armed.

4.8 Interrupt Enable Register

The bits in the Interrupt Enable Register are used to enable (or disable) the generation of interrupts based on the condition of certain circuit elements, known as interrupt sources. When a bit in this register associated with a given interrupt source is High, an interrupt will be generated by the rising edge of that source's Interrupt Status Register bit (See Section 4.9). This register is illustrated in Figure 4–8 and described in Table 4–9.

Figure 4–8: Interrupt Enable Register

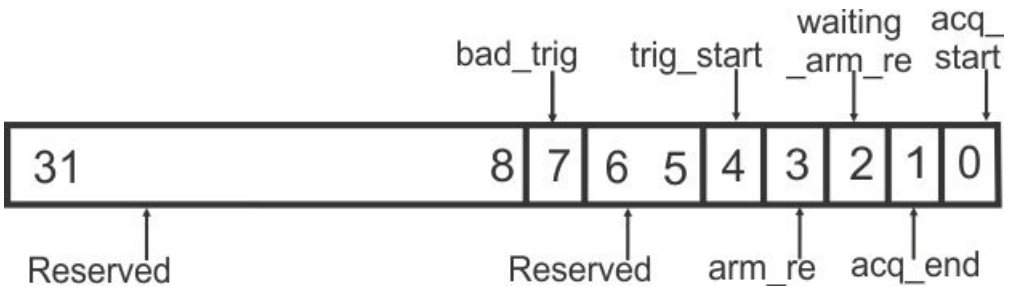


Table 4–9: Interrupt Enable Register (Base Address + 0x24)

Bits	Field Name	Default Value	Access Type	Description
31:8	Reserved	N/A	N/A	**Reserved**
7	bad_trig	0	R/W	**Bad Trigger:** This bit enables/ disables the Bad Trigger interrupt source. The Bad Trigger interrupt source indicates that a trigger occurred before the last acquisition was completed. 0 = Disable interrupt 1 = Enable interrupt
6:5	Reserved	–	–	**Reserved**
4	trig_start	0	R/W	**Start of Trigger:** This bit enables/ disables the Start of Trigger interrupt source. The Start of Trigger interrupt source indicates that the trigger has been enabled. 0 = Disable interrupt 1 = Enable interrupt

Table 4–9: Interrupt Enable Register (Base Address + 0x24) (Continued)

Bits	Field Name	Default Value	Access Type	Description
3	arm_re	0	R/W	**Arm was Reset:** This bit enables/ disables the Arm was Reset interrupt source. The Arm was Reset interrupt source indicates that the trigger arm was reset then re–enabled. 0 = Disable interrupt 1 = Enable interrupt
2	waiting_arm_re	0	R/W	**Waiting for Arm Start:** This bit enables/ disables the Waiting for Arm Start interrupt source. The Waiting for Arm Start interrupt source indicates that the trigger control state machine has transitioned into the Waiting for Arm state. 0 = Disable interrupt 1 = Enable interrupt
1	acq_end	0	R/W	**End of Acquisition:** This bit enables/ disables the End of Acquisition interrupt source. The End of Acquisition interrupt source indicates that the waveform started by the last trigger has completed. 0 = Disable interrupt 1 = Enable interrupt
0	acq_start	0	R/W	**Start of Acquisition:** This bit enables/ disables the Start of Acquisition interrupt source. The Start of Acquisition interrupt source indicates that a trigger has been reached. 0 = Disable interrupt 1 = Enable interrupt

4.9 Interrupt Status Register

The Interrupt Status Register has read–only access associated with each interrupt condition. A status bit changes to '1' when the source interrupt occurs. When a status bit in this register changes to '1' the corresponding flag bit in the Interrupt Flag Register is set to '1'. A status bit in this register clears to '0' when that interrupt condition clears, whereas the associated flag bit in the Interrupt Flag Register remains at logic '1' until it is explicitly cleared by the user.

Some of the interrupt sources are transient and so may not appear in the Interrupt Status Register at the time it is read. In such cases, use the Interrupt Flag Register to see the interrupt conditions that have occurred. The Interrupt Status Register is illustrated in Figure 4–9 and described in Table 4–10.

Figure 4–9: Interrupt Status Register

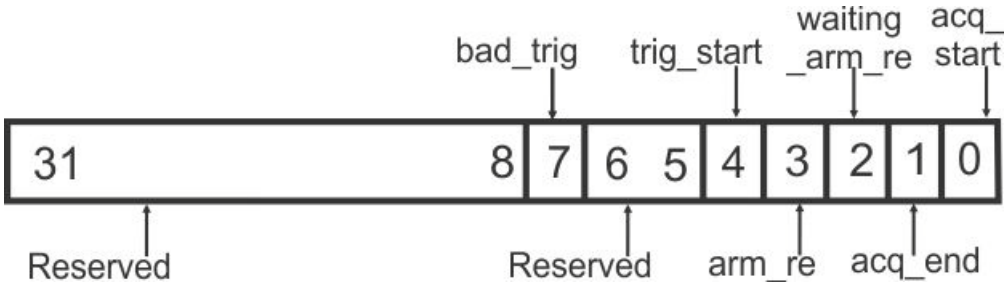


Table 4–10: Interrupt Status Register (Base Address + 0x28)

Bits	Field Name	Default Value	Access Type	Description
31:8	Reserved	N/A	N/A	**Reserved**
7	bad_trig	0	RO	**Bad Trigger:** This bit indicates the status of the Bad Trigger interrupt source. The Bad Trigger interrupt source indicates that a trigger occurred before the last acquisition was completed. 0 = No interrupt 1 = Interrupt condition asserted
6:5	Reserved	–	–	**Reserved**
4	trig_start	0	RO	**Start of Trigger:** This bit indicates the status of the Start of Trigger interrupt source. The Start of Trigger interrupt source indicates that the trigger has been enabled. 0 = No interrupt 1 = Interrupt condition asserted

Table 4–10: Interrupt Status Register (Base Address + 0x28) (Continued)

Bits	Field Name	Default Value	Access Type	Description
3	arm_re	0	RO	**Arm was Reset:** This bit indicates the status of the Arm was Reset interrupt source. The Arm was Reset interrupt source indicates that the trigger arm was reset then re–enabled. 0 = No interrupt 1 = Interrupt condition asserted
2	waiting_arm_re	0	RO	**Waiting for Arm Start:** This bit indicates the status of the Waiting for Arm Start interrupt source. The Waiting for Arm Start interrupt source indicates that the trigger control state machine has transitioned into the Waiting for Arm state. 0 = No interrupt 1 = Interrupt condition asserted
1	acq_end	0	RO	**End of Acquisition:** This bit indicates the status of the End of Acquisition interrupt source. The End of Acquisition interrupt source indicates that the waveform started by the last trigger has completed. 0 = No interrupt 1 = Interrupt condition asserted
0	acq_start	0	RO	**Start of Acquisition:** This bit indicates the status of the Start of Acquisition interrupt source. The Start of Acquisition interrupt source indicates that a trigger has been reached. 0 = No interrupt 1 = Interrupt condition asserted

4.10 Interrupt Flag Register

The Interrupt Flag Register has read/clear access associated with each interrupt condition. When reset, this register has all bits set to '0' (cleared). Each flag bit in this register latches an interrupt occurrence. A '1' in any flag bit in this register indicates that an interrupt has occurred.

Note that when any status bit in the Interrupt Status Register, changes to '1' the corresponding flag bit in this register will also be set to '1'. However, when a status bit in the Interrupt Status Register clears from '1' to '0', the corresponding latched flag bit in this register does not clear, but remains at '1'. To clear the flag bits, write '1's to the desired bits. The flags are not affected by the Interrupt Enable Register. The Interrupt Flag Register is illustrated in Figure 4–10 and described in Table 4–11.

Figure 4–10: Interrupt Flag Register

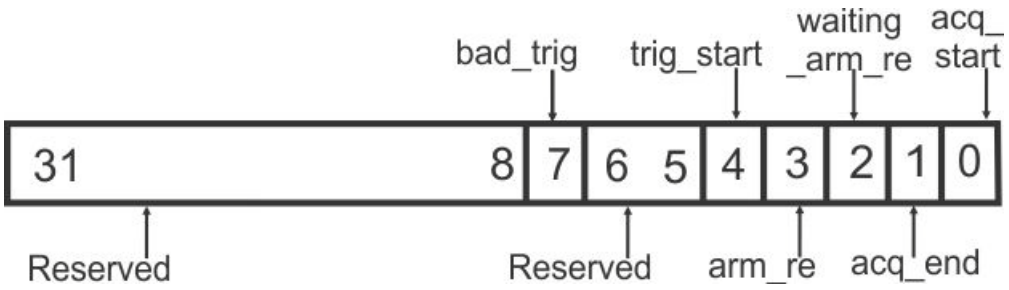


Table 4–11: Interrupt Flag Register (Base Address + 0x2C)

Bits	Field Name	Default Value	Access Type	Description
31:8	Reserved	N/A	N/A	**Reserved**
7	bad_trig	0	R/CLR	**Bad Trigger:** This bit indicates the status of the Bad Trigger interrupt flag. The Bad Trigger interrupt source indicates that a trigger occurred before the last acquisition was completed. **Read:** 0 = No interrupt 1 = Interrupt latched **Clear:** 1 = Clear latch
6:5	Reserved	–	–	**Reserved**

Table 4–11: Interrupt Flag Register (Base Address + 0x2C) (Continued)				
Bits	**Field Name**	**Default Value**	**Access Type**	**Description**
4	trig_start	0	R/CLR	**Start of Trigger:** This bit indicates the status of the Start of Trigger interrupt flag. The Start of Trigger interrupt source indicates that the trigger has been enabled. **Read:** 0 = No interrupt 1 = Interrupt latched **Clear:** 1 = Clear latch
3	arm_re	0	R/CLR	**Arm was Reset:** This bit indicates the status of the Arm was Reset interrupt flag. The Arm was Reset interrupt source indicates that the trigger arm was reset then re–enabled. **Read:** 0 = No interrupt 1 = Interrupt latched **Clear:** 1 = Clear latch
2	waiting_arm_re	0	R/CLR	**Waiting for Arm Start:** This bit indicates the status of the Waiting for Arm Start interrupt flag. The Waiting for Arm Start interrupt source indicates that the trigger control state machine has transitioned into the Waiting for Arm state. **Read:** 0 = No interrupt 1 = Interrupt latched **Clear:** 1 = Clear latch
1	acq_end	0	R/CLR	**End of Acquisition:** This bit indicates the status of the End of Acquisition interrupt flag. The End of Acquisition interrupt source indicates that the waveform started by the last trigger has completed. **Read:** 0 = No interrupt 1 = Interrupt latched **Clear:** 1 = Clear latch
0	acq_start	0	R/CLR	**Start of Acquisition:** This bit indicates the status of the Start of Acquisition interrupt flag. The Start of Acquisition interrupt source indicates that a trigger has been reached. **Read:** 0 = No interrupt 1 = Interrupt latched **Clear:** 1 = Clear latch

Chapter 5: Designing with the Core

This chapter includes guidelines and additional information to facilitate designing with the AXI4–Stream BRAM to Wave 256 Core.

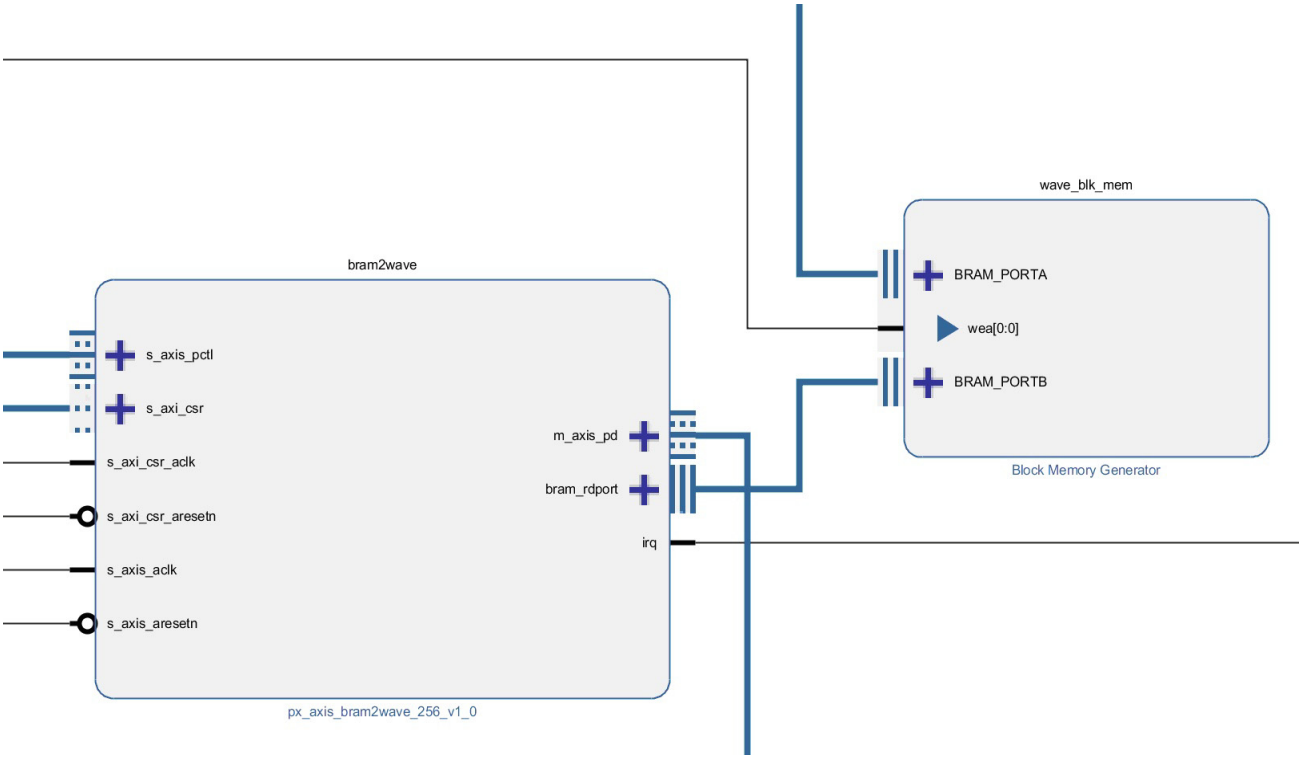
5.1 General Design Guidelines

The AXI4–Stream BRAM to Wave 256 Core, when combined with the required External BRAM, provides the logic necessary to generate an AXI4–Stream waveform from a wave table stored in the BRAM. The user can customize the core by setting the generic parameter based on the application requirement as described in Section 2.5.

5.2 Generating the External BRAM

The External BRAM is a Xilinx core which must be generated as part of the design process and added to the top–level design alongside the AXI4–Stream BRAM to Wave Core as shown in Figure 5.1. Details on generating the core can be found in Section 6.3.

Figure 5–1: AXI4–Stream BRAM to Wave 256 Core with External BRAM



5.3 Clocking

AXI4−Lite Clock: `s_axi_csr_aclk`.

This clock is used to clock the AXI4−Lite Control/Status Register (**`s_axi_csr`**) interface of the core and its associated logic.

AXI4−Stream Clock: `s_axis_aclk`.

This clock provides clocking for all of the waveform generation logic, the BRAM inter−face, and the AXI4−Stream interfaces.

5.4 Resets

CSR Reset: `s_axi_csr_aresetn`.

This is an active−low synchronous reset associated with the **`s_axi_csr_aclk`**. When asserted, all CSR state machines in the core are reset and the registers are returned to their initial values.

AXI4−Stream Reset: `s_axis_aresetn`.

This is an active−low synchronous reset associated with the **`s_axis_aclk`**. When asserted the AXI4−stream interfaces are reset.

5.5 Interrupts

This core has an edge−type (rising edge−triggered) interrupt output. It is synchronous with the **`s_axi_csr_aclk`**. On the rising edge of any interrupt signal, a one clock cycle wide pulse is output from the core on its **`irq`** output. Each interrupt event is stored in two registers, accessible on the **`s_axi_csr`** bus. The Interrupt Status Register always reflects the current state of the interrupt condition, which may have changed since the generation of the interrupt. The Interrupt Flag Register latches the occurrence of each interrupt, in a bit that retains its state until explicitly cleared.The interrupt flags can be cleared by writing '1' to the associated bit's location. All interrupt sources that are enabled (via the Interrupt Enable Register) are "OR ed" onto the **`irq`** output.

NOTE: All interrupt sources are latched in the Interrupt Flag Register, even when an interrupt source is not enabled (via the Interrupt Enable Register).

NOTE: Because this core uses edge−triggered interrupts, the fact that an interrupt condition may remain active after servicing will not cause the generation of a new interrupt. A new interrupt will only be generated by another rising edge on an interrupt source.

5.6 Interface Operation

AXI4–Lite Slave CSR Interface: This is the control/status/interrupt register interface. It is associated with the **`s_axi_csr_aclk`**, and is a standard AXI4–Lite type interface. See Chapter 4 for the control register memory map and for more details on the registers that can be accessed through this interface.

PCTL AXI4–Stream Slave Interface: This is the interface which carries the Gate/Trigger from the Sync Bus to the core. For more details about this interface refer to Section 3.2.1.

AXI4–Stream Master Interface: This is the output data interface which carries the generated waveform data out of the core. For more details about this interface refer to Section 3.2.2.

External BRAM Interface: This is the interface for the external BRAM's read port. For more details about this interface refer to Section 3.3.

5.7 Programming Sequence

This section briefly describes the programming sequence for the AXI4–Stream BRAM to Wave 256 Core.

1) Ensure that the Interrupt Flag Register is cleared.

2) Enable the Interrupt Enable Register bits based on the user design requirement.

3) Write the desired values to the Control Registers.

4) Observe waveform output data when the trigger condition is met.

5.8 Timing Diagrams

The timing diagram for the AXI4–Stream BRAM to Wave 256 Core, shown in Figure 6–8, is obtained by running the simulation of the test bench of the core in Vivado VSim environment. For more details about the test bench, refer to Section 6.6.

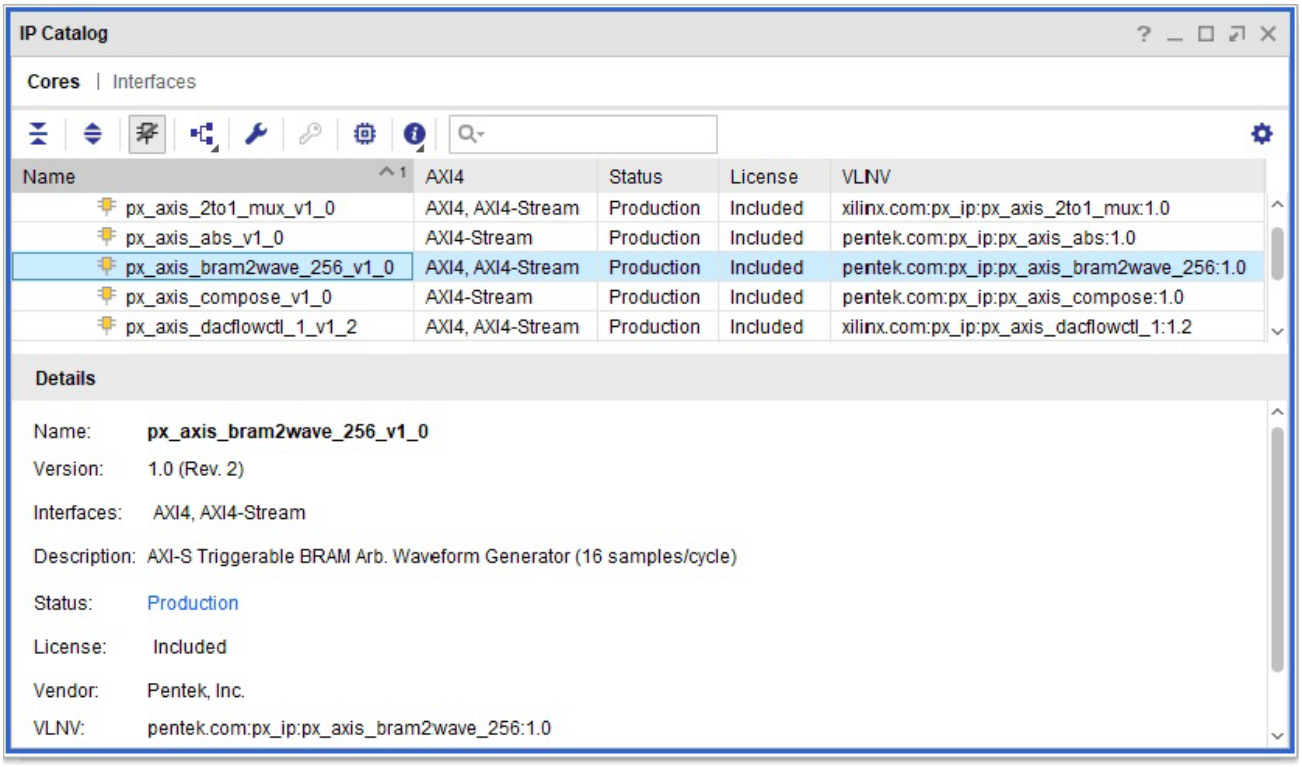
This page is intentionally blank

Chapter 6: Design Flow Steps

6.1 Pentek IP Catalog

This chapter describes customization and generation of the Pentek AXI4–Stream BRAM to Wave 256 Core. It also includes simulation, synthesis, and implementation steps that are specific to this IP core. This core can be generated from the Vivado IP Catalog when the Pentek IP Repository has been installed. It will appear in the IP Catalog list as **px_axis_bram2wave_256_v1_0** as shown in Figure 6–1.

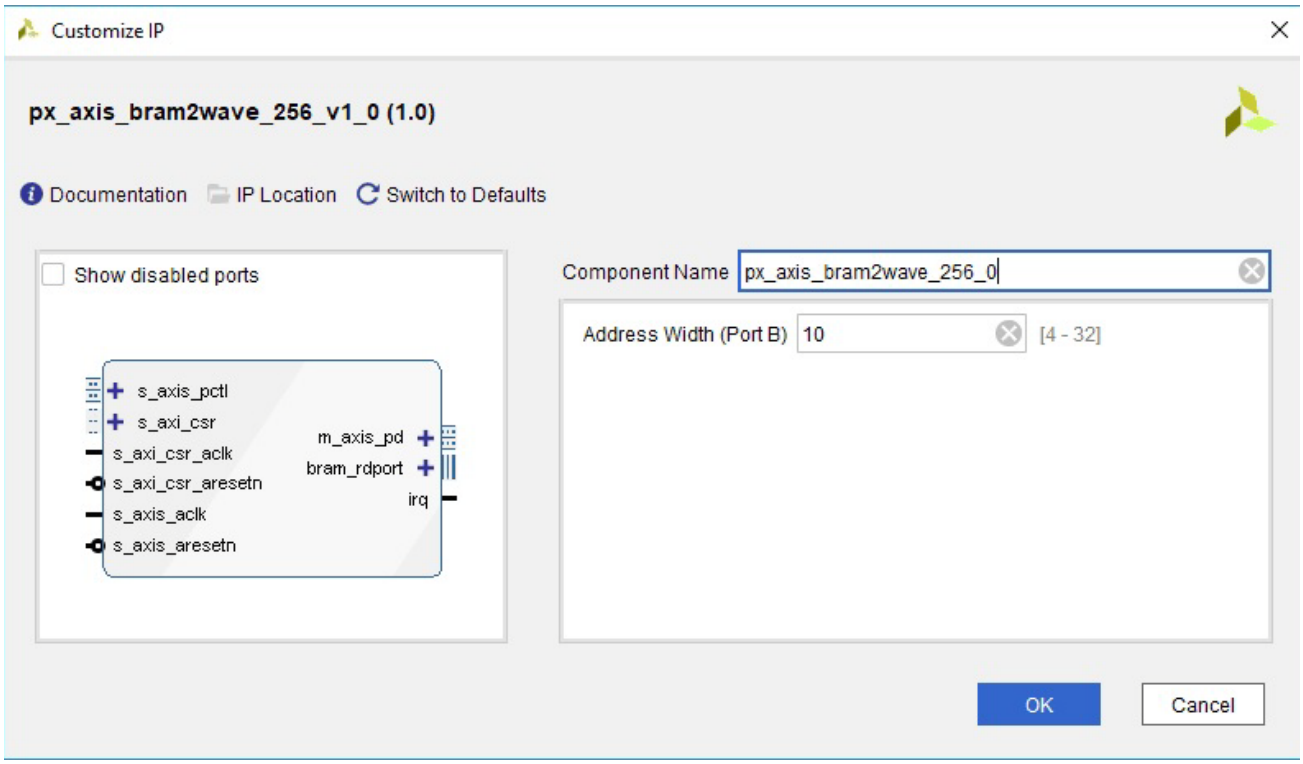
Figure 6–1: AXI4–Stream BRAM to Wave 256 Core in Pentek IP Catalog



6.1 Pentek IP Catalog (continued)

When you select the **px_axis_bram2wave_256_v1_0** core, a screen appears that shows the core's symbol and the core's parameters (see Figure 6–2). The core's symbol is the box on the left side.

Figure 6–2: AXI4–Stream BRAM to Wave 256 Core Symbol



6.2 User Parameters

There is one user parameter for this IP core. It is described in Section 2.5.

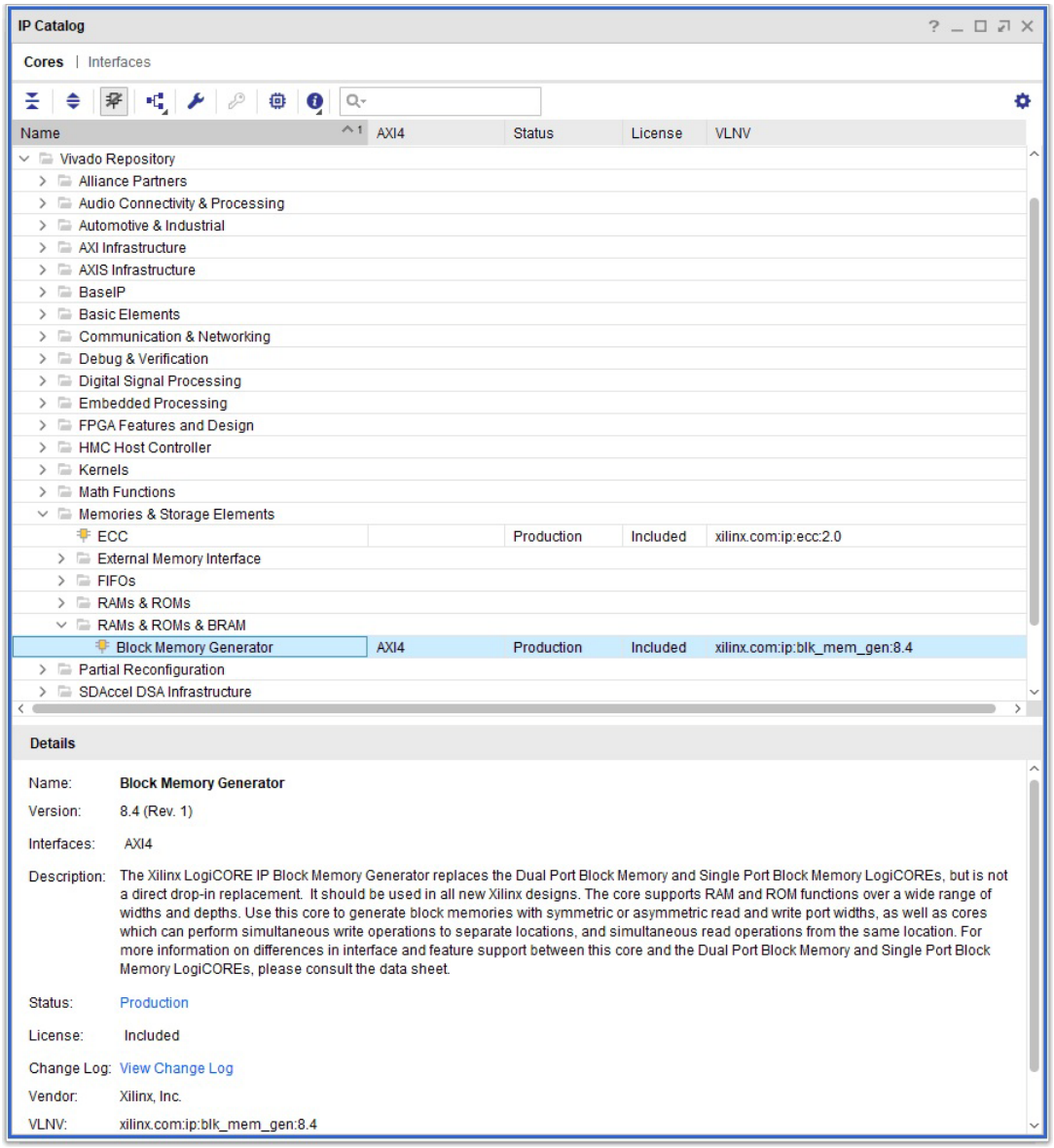
6.3 Generating the external Xilinx BRAM

NOTE: The parameters shown in this section are for demonstrative purposes only, and may not be optimal for the user's design. The user should choose settings based on the requirements of the system being designed.

6.3.1 Step 1

Add the core to the top level block diagram by selecting Vivado Repository => Memories & Storage Elements => RAMs & ROMs & BRAM => Block Memory Generator (see Figure 6–3).

Figure 6–3: Block Memory Generator in the Vivado IP Catalog

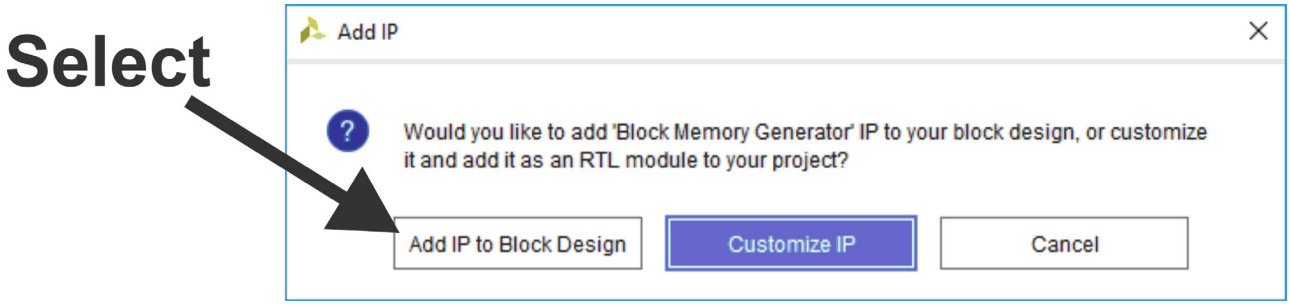


6.3 Generating the external Xilinx BRAM

6.3.2 Step 2

In the "Add IP" dialog box, select "Add IP to Block Design" (see Figure 6–4).

Figure 6–4: "Add IP" Dialog Box for Xilinx FIFO Generator

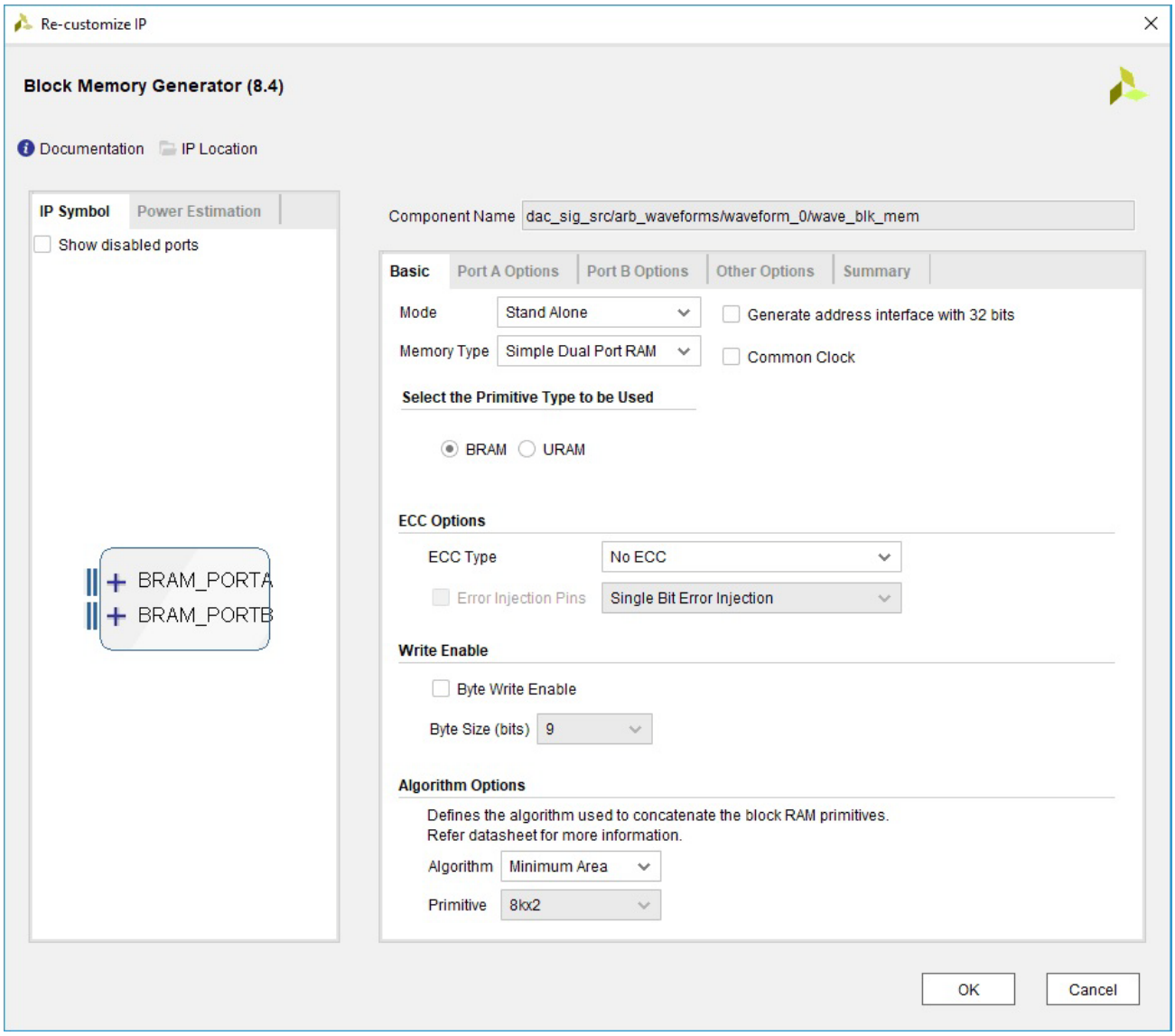


6.3 Generating the external Xilinx BRAM

6.3.3 Step 3

Double–click on the Block Memory Generator block in the Block Diagram to open the customization dialog box. In the "Basic" tab, set the parameters as required. See the example settings in Figure 6–5.

Figure 6–5: "Basic" Tab Setup for Xilinx BRAM Generator

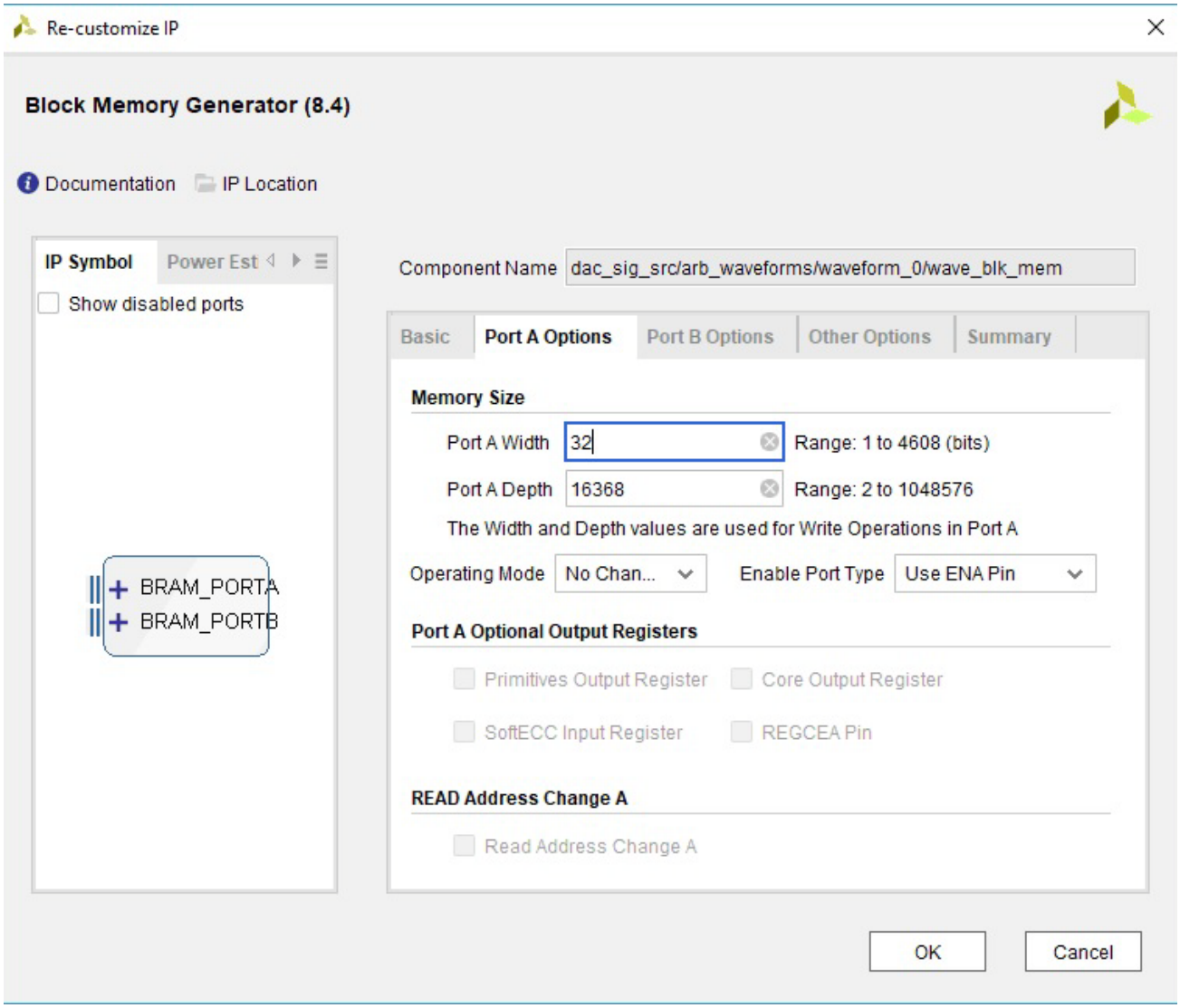


6.3 Generating the external Xilinx BRAM

6.3.4 Step 4

In the "Port A Options" tab, set the parameters as required. See the example settings in Figure 6–6.

Figure 6–6: "Port A Options" Tab Setup for Xilinx BRAM Generator

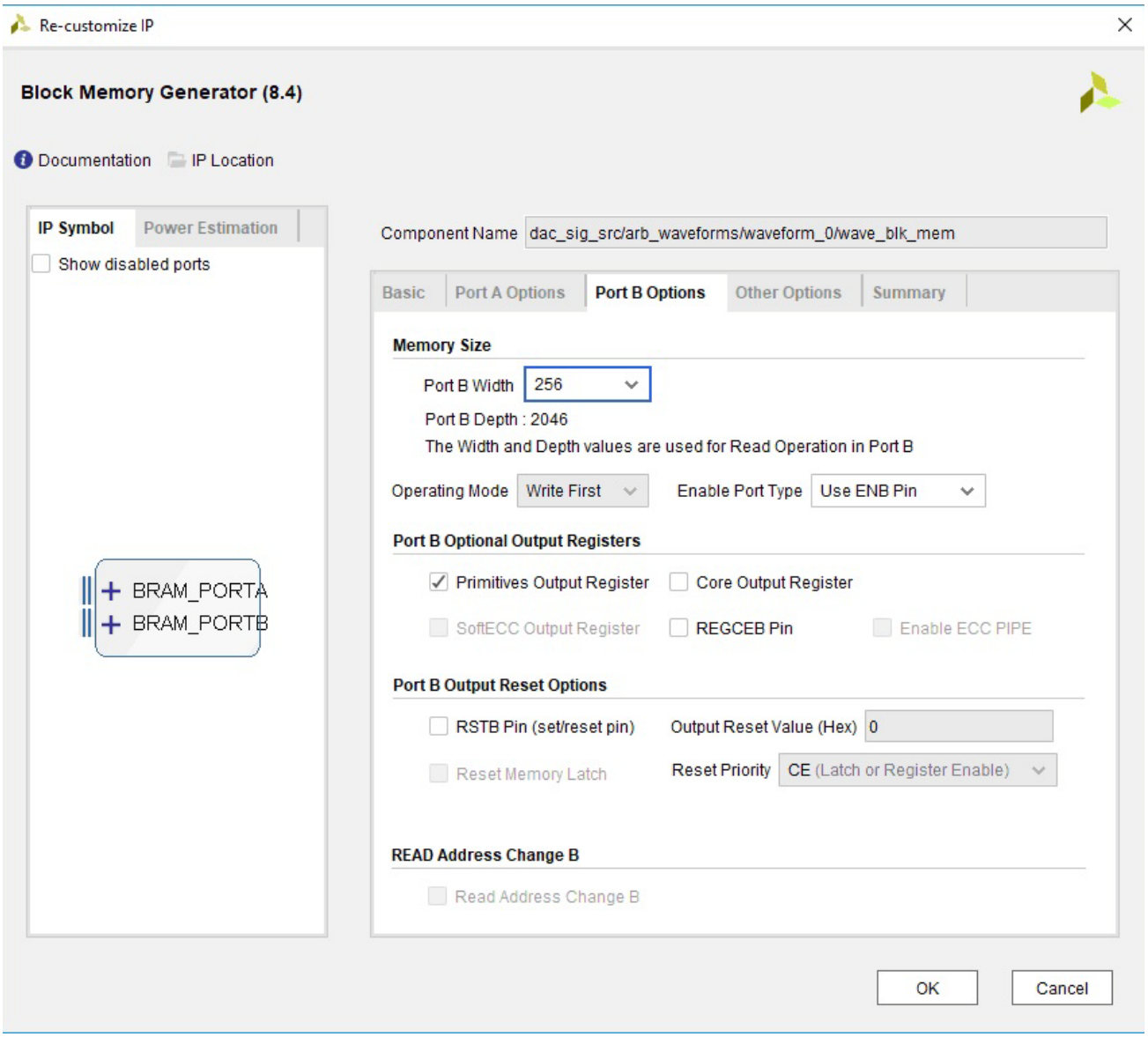


6.3 Generating the external Xilinx BRAM

6.3.5 Step 5

In the "Port B Options" tab, set the parameters as required. See the example settings in Figure 6–7. No changes in the "Other Options" tab are necessary. Select the "OK" button in the lower right corner of the dialog box to complete the setup.

Figure 6–7: "Port B Options" Tab Setup for Xilinx BRAM Generator



6.4 Generating Output

For more details about generating and using IP in the Vivado Design Suite, refer to the *Vivado Design Suite User Guide – Designing with IP*.

6.5 Constraining the Core

This section contains information about constraining the core in Vivado Design Suite environment.

Required Constraints

The XDC constraints for this core are not provided with this core. The necessary con−straints can be applied at the top level module of the user design.

Device, Package, and Speed Grade Selections

This IP works for Zynq Ultrascale+ RFSOC FPGAs.

Clock Frequencies

For the streaming data path clock (**`s_axis_aclk`**), the maximum clock frequency is 500 MHz.

The AXI4−Lite interface clock (**`s_axi_csr_aclk`**) frequency is 250 MHz.

Clock Management

This section is not applicable for this IP core.

Clock Placement

This section is not applicable for this IP core.

Banking and Placement

This section is not applicable for this IP core.

Transceiver Placement

This section is not applicable for this IP core.

I/O Standard and Placement

This section is not applicable for this IP core.

6.6 Simulation

The AXI4–Stream BRAM to Wave 256 Core has a test bench which generates the output waveforms using the Vivado VSim environment. The test bench is designed to run with the following parameters:

1) AXI4–Stream clock (**s_axis_aclk**) frequency: 400 MHz

2) AXI4–Lite CSR clock (**s_axis_csr_aclk**) frequency: 250 MHz

3) Parameter "**addr_width**" is set to 10 (default)

The register settings for the simulation are contained in a setup file called **test_parameters.txt**, which can be found in the top–level directory for the core. The contents of the **test_parameters.txt** file along with descriptions of the parameters are provided in Table 6–1.

Table 6-1: Test Parameters File Contents and Parameter Descriptions

Parameter	Type	Value	Description
MODE_SEL	std_ulogic _vector [3:0]	0x1	**Mode Select:** This parameter sets the trigger mode as follows: 0x0 = Gate mode 0x1 = Trigger mode 0x2 = Trigger Hold mode 0x3 = Reseved
DATA_MODE_SEL	std_ulogic _vector [3:0]	0x1	**Data Mode Select:** This parameter sets the data mode as follows: 0x0 = Time Packed Real Data (I[t1],I[t0]) 0x1 = Channel Packed I/Q Data (Q0, I0) or two channel Real Data (Ib, Ia). 0x2 = Reserved 0x3 = Reserved
STAY_ARMED	Boolean	True	**Stay Armed:** This parameter determines the end–of–trigger action as follows: True = Stay armed at end of triggered gate False = Disarm trigger at end of triggered gate
TRIG_SPACE	std_ulogic _vector [31:0]	0x00001000	**Trigger Space:** This parameter controls the space between triggers (in **s_axis_aclk** periods).
BRAM_START _ADDRESS		0x00000000	**BRAM Start Address:** This is the address of the first data word in the waveform table.
BRAM_END_ADDRESS		0x0000001F	**BRAM End Address:** Address of the last data word in the waveform table.
TRIGGER_DLY_VALUE		0x00000000	**Trigger Delay Value:** This is the value used to load to the delay counter.
TRIGGER_LENGTH _VALUE		0x00000400	**Trigger Length Value:** This is the value used to load to the sample counter.

6.6 Simulation (continued)

Once the reset (**`s_axis_aresetn`**) is released, the testbench begins setting the registers with the values specified in the **`test_parameters.txt`** file. It then arms the trigger (an interrupt is generated) and begins writing waveform table data to the external BRAM. Note that for this simulation a simple counter pattern is loaded instead of an actual waveform table.

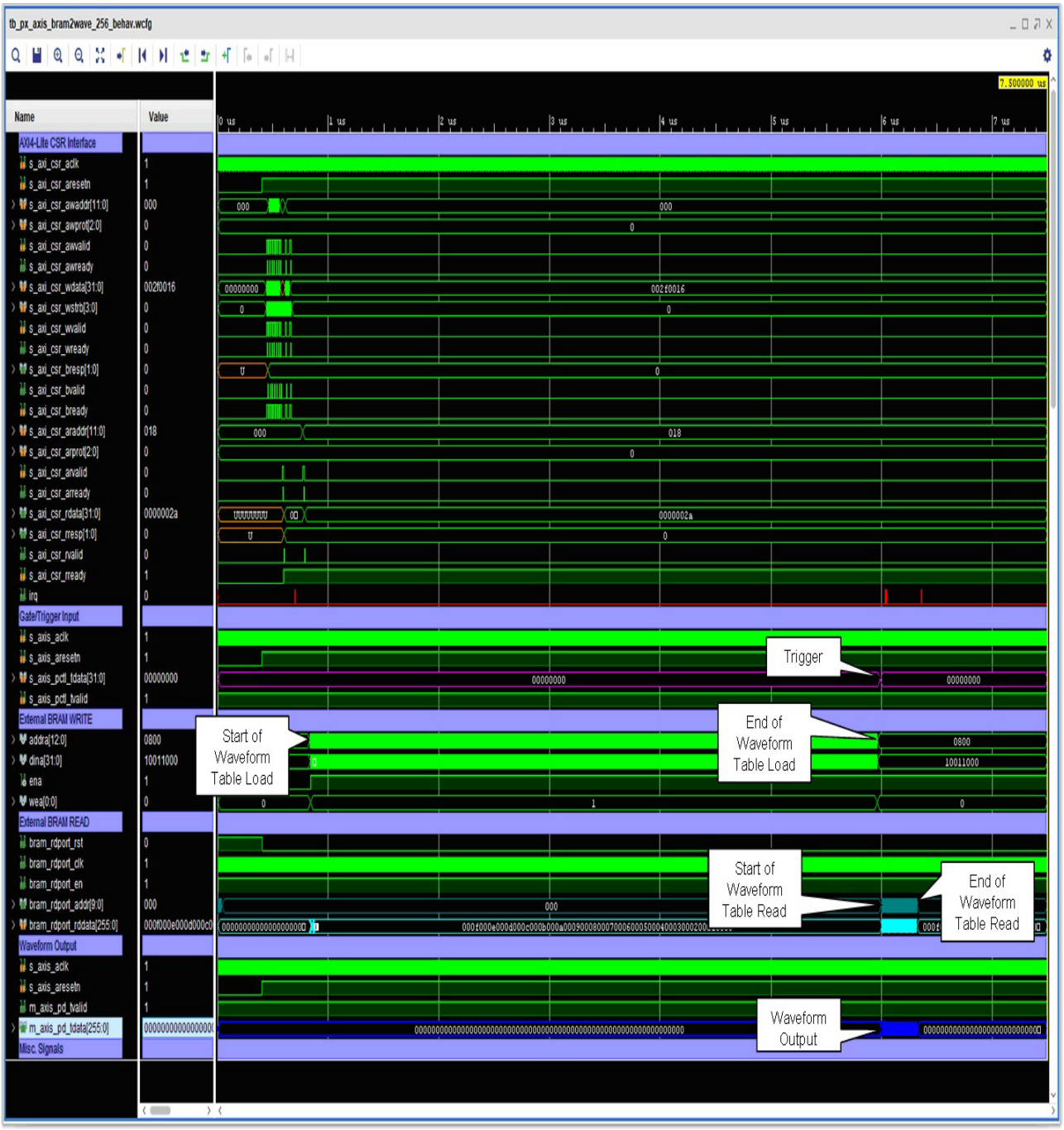
Once the waveform table is loaded, the testbench forces a trigger on the PCTL AXI4–Stream Slave input bus. This causes an interrupt to be generated and wavetable data begins to appear on the AXI4–Stream Master output bus.

When the last waveform data appears on the output, an interrupt is generated.

The signals presented in the waveform window demonstrate this to the user as shown in Figure 6–8.

6.6 Simulation (continued)

Figure 6-8: AXI4-Stream BRAM to Wave 256 Core Test Bench Simulation Output



6.7 Synthesis and Implementation

For details about synthesis and implementation see the *Vivado Design Suite User Guide – Designing with IP*.