IP CORE MANUAL

AXI4–Stream Chirp Generator IP

px_axis_chirp_gen

PENTEK

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

Manual Part Number: 807.48420 Rev: 1.2 – November 1, 2018

Manual Revision History

Date	Version	Comments
9/18/18	1.0	Initial Release
10/19/18	1.1	Minor revisions. Revised Table 3–1, Table 4–2, Sect 4.2, Table 4–8, Table 4–9, and Sect 6.5.
11/1/18	1.2	Revised Table 4–2 and Sect 6.5.

Legal Notices

Copyright

Trademarks

Printed in the United States of America.

Table of Contents

Table of Contents

Page

Chapter 4: Register Space

Chapter 5: Designing with the Core

Table of Contents

Page

Chapter 6: Design Flow Steps

Table of Contents

Page

This page is intentionally blank

IP Facts

Description

Pentek's Navigator™ The Pentek Chirp Generator Core generates a user–configurable chirp signal in the form of an AXI4–Stream PDTI–Style data stream which can be synchronized to an incoming AXI4–Stream timestamp data stream.

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 Chirp Generator Core.

Features

- User configurable frequency, phase offset, frequency ramp rate and pulse width
- Generates up to 8 samples per cycle – user configurable
- User selectable trigger from the incoming stream
- Interrupts available for trigger armed, start and end events
- Register access through AXI4–Lite CSR interface

Table 1–1: IP Facts Table

Core Specifics	
Supported Design Family[a]	Kintex® Ultrascale
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 2018.2 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

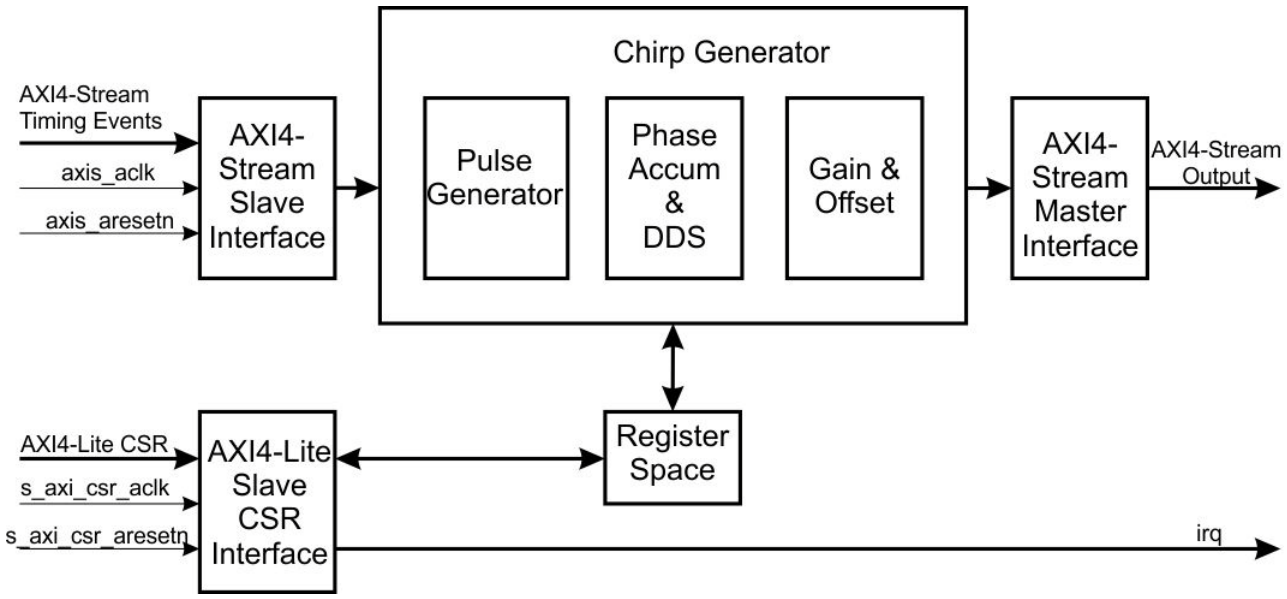
1.1 Functional Description

The Pentek Chirp Generator Core accepts input Timing Event Streams (Sample Clock, Reset and Timing Signals) through an AXI4–Stream Slave Interface and delivers a user configurable chirp output signal stream through a PDTI–Style AXI4–Stream Master Interface. For details on the AXI4–Stream interfaces, refer to Section 3.2. Parameter registers for chirp generation and for trigger settings are accessible via an AXI4–Lite CSR interface. For register map details see Chapter 4. For details on the AXI4–Lite CSR interface, refer to Section 3.1.

Latency for the Pentek Chirp Generator Core is 29 (**`s_axis_ptctl_tvalid-qualified`**) clock cycles. Backpressure on the AXI4–Stream interfaces is not supported.

Figure 1–1 is a top–level block diagram of the AXI4–Stream Chirp Generator Core. The modules within the block diagram are explained in other sections of this manual.

Figure 1–1: AXI4–Stream Chirp Generator Core Block Diagram



1.1 Functional Description (continued)

- **AXI4−Stream Slave Interface:** This module implements an AXI4−Stream Slave interface for the input timing events data stream for the Core. For additional details about the AXI4−Stream Slave Interface, refer to **Section 3.2 AXI4−Stream Core Interfaces**.

- **Chirp Generator:** This module generates the chirp signal based on user−defined parameters, and is composed of the following sub−blocks:

    - Pulse Generator: Sets−up the triggering and timing for the Chirp pulses.
    - Phase Accumulator and DDS: Generates the chirp signal based on the user−defined frequency, ramp and phase requirements.
    - Gain & Offset: Adjusts the gain and offset of the generated chirp signal as required.

- **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 AXI4−Stream Core Interfaces**.

- **AXI4−Lite Interface:** This module implements a 32−bit AXI4−Lite Slave Interface to access the register space. For more details about the AXI4−Lite Interface, refer to Section 3.1 AXI4−Lite Core Interfaces.

- **AXI4−Stream Interfaces:** The AXI4−Stream Chirp Generator Core has AXI4−Stream Interfaces to transfer requests and receive responses to/from the DDR4 SDRAM Memory Controller IP Core. This core also has AXI4−Stream Interfaces to receive timing event data streams and also to transfer the generated output AXI data streams. For more details about the AXI4−Stream Interfaces, refer to **Section 3.2 AXI4−Stream Core Interfaces**.

- **Register Space:** This module contains the control registers for the core. The registers are accessed through the AXI4−Lite CSR interface. For register map details see **Chapter 4**.

1.2 Applications

The Chirp Generator Core can be used for generating a user−configurable PDTI−Style AXI4 Chirp Signal Stream and can be incorporated into any Kintex Ultrascale FPGA.

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

This page is intentionally blank

Chapter 2: General Product Specifications

2.1 Standards

The AXI4–Stream Chirp Generator 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 Chirp Generator Core is limited only 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

This core is designed to meet a target frequency of 250 MHz on a Kintex Ultrascale –2 speed grade FPGA. 250 MHz is typically the PCI Express® (PCIe®) AXI Bus clock frequency.

2.3 Resource Utilization

The resource utilization of the AXI4–Stream Chirp Generator Core is shown in Table 2–1. Resources have been estimated for the Kintex Ultrascale XCKU060 –2 speed grade device. These values were generated using the Vivado Design Suite.

Table 2–1: Resource Usage and Availability

Resource	# Used
LUTs	1458
Flip–Flops	4014
DSP48E2s	7

NOTE: Actual utilization may vary based on the user design in which the AXI4–Stream Chirp Generator Core is incorporated.

2.4 Limitations and Unsupported Features

This core does not support backpressure on the AXI4–Stream interfaces.

2.5 Generic Parameters

The generic parameters of the AXI4–Stream Chirp Generator Core are described in Table 2–2. These parameters can be set as required by the user application while customizing the core.

Table 2-2: Generic Parameters

Port/Signal Name	Type	Description
samples_per_cycle	Integer	**Samples per Cycle:** This parameter defines the data rate of the outgoing data stream. Valid range is 1 to 8. The default setting for this parameter is 8.
interleave_output	Boolean	**Interleave Output:** This parameter is used to indicate whether the outgoing stream has the i and q data interleaved onto the same bus or are output on separate busses. The default setting for this parameter is TRUE.

Chapter 3: Port Descriptions

This chapter provides port descriptions for the following interface types:

- AXI4–Lite Core Interfaces
- AXI4–Stream Core Interfaces

3.1 AXI4–Lite Core Interfaces

The Pentek Chirp Generator Core uses the Control/Status Register (CSR) interface to access the control, status and interrupt registers 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, status and interrupt registers in the Pentek Chirp Generator 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_aclk	Input	1	**Clock**
s_axi_aresetn	Input	1	**Reset:** Active low. This will reset the state machine within the core.
s_axi_csr_awaddr	Input	5	**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 Chirp Generator Core.
s_axi_csr_awprot	Input	3	**Protection:** The Chirp Generator 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 Chirp Generator 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**.
s_axi_csr_awready	Output	1	**Write Address Ready:** This output is asserted by the Chirp Generator 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.

Table 3-1: Control/Status Register (CSR) Interface Port Descriptions (Continued)			
Port	**Direction**	**Width**	**Description**
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 Chirp Generator 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 Chirp Generator 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 Chirp Generator 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.
s_axi_csr_araddr	Input	5	**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 Chirp Generator Core.
s_axi_csr_arprot	Input	3	**Protection:** These bits are ignored by the Chirp Generator Core.

Table 3-1: Control/Status Register (CSR) Interface Port Descriptions (Continued)			
Port	**Direction**	**Width**	**Description**
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 Chirp Generator 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 Chirp Generator 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 Chirp Generator 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 Pentek Chirp Generator Core has the following AXI4–Stream Interfaces, which are used to transfer data streams.

3.2.1 Stream Data Interface

The input (slave) interface is an AXI4–Stream interface containing timestamp and trigger/gate information. The output (master) interface is a PDTI–Style interface which will contain the generated waveform when the user–defined trigger conditions are met on the input (slave) interface. Table 3–2 defines the ports in the Stream Data Interfaces. See the *AMBA AXI4–Stream Specification* for more details on the operation of the AXI4–Stream Interface.

Table 3-2: Stream Data Interface Port Descriptions

Port	Direction	Width	Description
s_axis_aclk	Input	1	**AXI4–Stream Clock**
s_axis_aresetn	Input	1	**Reset:** Active Low.
AXI4–Stream Slave Interfaces			
s_axis_ptctl_tdata	Input	See Description	**Input Data** Width = (samples_per_cycle*3)+8
s_axis_ptctl_tvalid	Input	1	**Input Data Valid:** This signal is asserted by the user logic when data is valid on the **s_axis_pdti_tdata** bus.
AXI4–Stream Master Interface (Interleaved) **(**available only when **interleave_output** = TRUE**)**			
m_axis_pd_tdata	Output	See Description	**Output Data** Width = samples_per_cycle*32
m_axis_pd_tvalid	Output	1	**Output Data Valid:** This signal is asserted when data is valid on **m_axis_pd_tdata** bus.
AXI4–Stream Master Interface (Non–interleaved, I Data) (available only when **interleave_output** = FALSE)			
m_axis_pd_i_tdata	Output	See Description	**Output I Data** Width = samples_per_cycle*16
m_axis_pd_i_tvalid	Output	1	**Output I Data Valid:** This signal is asserted when data is valid on **m_axis_pd_i_tdata** bus.
AXI4–Stream Master Interface (Non–interleaved, Q Data) (available only when **interleave_output** = FALSE)			
m_axis_pd_q_tdata	Output	See Description	**Output Q Data** Width = samples_per_cycle*16
m_axis_pd_q_tvalid	Output	1	**Output Q Data Valid:** This signal is asserted when data is valid on **m_axis_pd_q_tdata** bus.

Chapter 4: Register Space

This chapter provides the memory map and register descriptions for the Register Space of the AXI4–Stream Chirp Generator 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 Register	0x00	R/W	Controls gain, reset, mode, arming and trigger select for the core.
Frequency Value Register	0x04	R/W	Initial Frequency Value – start phase increment value.
Phase Offset Register	0x08	R/W	Initial phase offset value
Frequency Ramp Rate Register	0x0C	R/W	Rate of change per sample period of the phase increment
Pulse Width Register	0x10	R/W	Length of triggered pulse in samples
Interrupt Enable Register	0x14	R/W	Interrupt Enable Register
Interrupt Status Register	0x18	RO	Interrupt Status Register
Interrupt Flag Register	0x1C	R/CLR	Interrupt Flag Register

4.1 Mode Control Register

This register controls the gain, reset, arming, and triggering of the core. The Mode Control Register is illustrated in Figure 4–1 and described in Table 4–2.

Figure 4–1: Mode Control Register

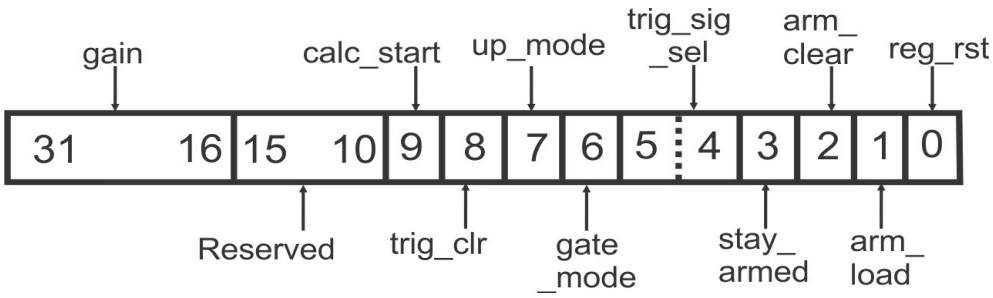


Table 4–2: Mode Control Register (Base Address + 0x00)

Bits	Field Name	Default Value	Access Type	Description
31:16	Gain	0x0400	R/W	**Gain:** Sets the overall gain of the core. 0x8000 = Unity Gain
15:10	Reserved	–	–	**Reserved**
9	calc_start	0	R/W	**Calculation Start:** When **`samples_per_cycle`** is > 1, setting this bit HIGH enables the calculation engine for the waveform generator. **NOTE:** To insure correct desired operation make sure ALL parameters are set to their respective desired values before enabling the calculation engine.
8	trig_clr	0	R/W	**Trigger Clear:** When HIGH the trigger is cleared.
7	up_mode	0	R/W	**Up Mode:** When HIGH the phase change of the output waveform is positive (sweep up), otherwise it is negative.
6	gate_mode	0	R/W	**Gate Mode:** When HIGH, GATE mode is selected. When LOW, TRIGGER mode is selected.
5:4	trig_sig_sel	00	R/W	**Trigger Signal Select:** Selects the trigger source as follows: 00 = Gate Positions 01 = Sync Positions 10 = PPS Positions 11 = Reserved
3	stay_armed	0	R/W	**Stay Armed:** When HIGH this bit prevents the trigger from being disarmed.
2	arm_clear	0	R/W	**Arm Clear:** When HIGH with the **stay_armed** bit LOW, this bit disarms the trigger.

Table 4–2: Mode Control Register (Base Address + 0x00) (Continued)				
Bits	**Field Name**	**Default Value**	**Access Type**	**Description**
1	arm_load	0	R/W	**Arm Load:** When HIGH, this bit arms the trigger.
0	reg_rst	0	R/W	**Register Reset:** When HIGH this bit resets the core's Trigger Arm state machine.

4.2 Frequency Value Register

The Frequency Value Register is used to set the starting phase increment value. The frequency value can be determined by using the following formula:

Frequency = ((sample rate (Hz)) * (Frequency Value[31:0]))/2^{32}

This register is illustrated in Figure 4–2 and described in Table 4–3.

Figure 4–2: Frequency Value Register

Table 4–3: Frequency Value Register (Base Address + 0x04)				
Bits	**Field Name**	**Default Value**	**Access Type**	**Description**
31:0	start_frq_incr	0x00000000	R/W	**Frequency Value:** Starting phase increment value

4.3 Phase Offset Register

The Phase Offset Register is used to set the initial phase offset value as follows:

Phase Offset (in radians) = $(2*\pi* \text{PhaseOffset}[31:0])/2^{32}$

This register is illustrated in Figure 4–3 and described in Table 4–4.

Figure 4–3: Phase Offset Register

Table 4–4: Phase Offset Register (Base Address + 0x08)

Bits	Field Name	Default Value	Access Type	Description
31:0	phase_offset	0x00000000	R/W	**Phase Offset:** Starting phase offset value

4.4 Frequency Ramp Rate Register

The Frequency Ramp Rate Register is used to set the rate of change per sample period of the phase increment as follows:

Frequency Ramp Rate (Hz/cycle) = ((sample rate (Hz) * (Frequency Ramp Rate[31:0])) / 2^{32}

This register is illustrated in Figure 4–4 and described in Table 4–5.

Figure 4–4: Frequency Ramp Rate Register

Table 4–5: Frequency Ramp Rate Register (Base Address + 0x0C)

Bits	Field Name	Default Value	Access Type	Description
31:0	rate	0x00000000	R/W	**Ramp Rate:** Sets the rate of change per sample period of the phase increment.

4.5 Pulse Width Register

The Pulse Width Register is used to set the Length of a triggered pulse in samples as follows:

Pulse length in clock cycles = Pulse Width[31:0]/samples–per–cycle (Must be a multiple of the samples–per–cycle value or it will be rounded down)

NOTE: If Pulse width is set to 0x00000000, once triggered the output stream will continue indefinitely until cleared.

This register is illustrated in Figure 4–5 and described in Figure 4–5.

Figure 4–5: Pulse Width Register

Table 4–6: Pulse Width Register (Base Address + 0x10)

Bits	Field Name	Default Value	Access Type	Description
31:0	pulse_width	0x00000000	R/W	**Pulse Width:** Sets the length of the triggered pulse in samples.

4.6 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 4.7). This register is illustrated in Figure 4–6 and described in Table 4–7.

Figure 4–6: Interrupt Enable Register

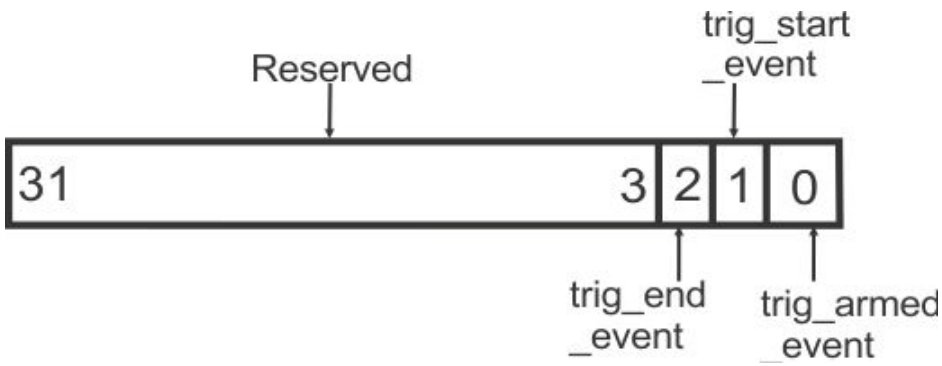


Table 4–7: Interrupt Enable Register (Base Address + 0x14)

Bits	Field Name	Default Value	Access Type	Description
31:3	Reserved	N/A	N/A	**Reserved**
2	trig_end _event	0	R/W	**Trigger End Event Interrupt:** This bit enables/ disables the trigger end event interrupt source. The trigger end event source indicates that the trigger event has ended. 0 = Disable interrupt 1 = Enable interrupt
1	trig_start _event	0	R/W	**Trigger Start Event Interrupt:** This bit enables/ disables the trigger start event interrupt source. The trigger start event source indicates that a trigger event has started. 0 = Disable interrupt 1 = Enable interrupt
0	trig_armed _event	0	R/W	**Trigger Armed Event Interrupt:** This bit enables/ disables the trigger armed event interrupt source. The trigger armed event source indicates that a trigger condition has been set and generation of the waveform output will begin when the trigger condition has been met. 0 = Disable interrupt 1 = Enable interrupt

4.7 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 also 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. This register is illustrated in Figure 4–7 and described in Table 4–8.

Figure 4–7: Interrupt Status Register

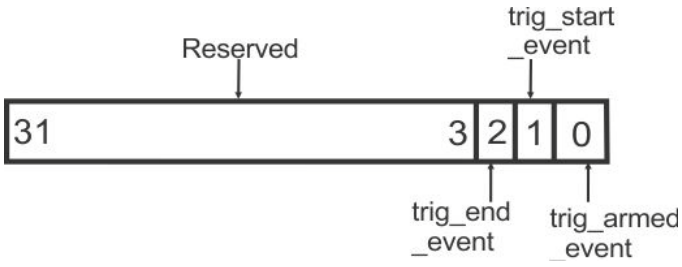


Table 4–8: Interrupt Status Register (Base Address + 0x18)

Bits	Field Name	Default Value	Access Type	Description
31:3	Reserved	N/A	N/A	**Reserved**
2	trig_end _event	0	RO	**Trigger End Event Interrupt:** This bit reports the current status of the trigger end event interrupt. The trigger end event source indicates that the trigger event has ended. 0 = No interrupt 1 = Interrupt condition asserted
1	trig_start _event	0	RO	**Trigger Start Event Interrupt:** This bit reports the current status of the trigger start event interrupt. The trigger start event source indicates that a trigger event has started. 0 = No interrupt 1 = Interrupt condition asserted
0	trig_armed _event	0	RO	**Trigger Armed Event Interrupt:** This bit reports the current status of the trigger armed event interrupt. The trigger armed event interrupt indicates that a trigger condition has been set and generation of the waveform output will begin when the trigger condition has been met. 0 = No interrupt 1 = Interrupt condition asserted

4.8 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. This register is illustrated in Figure 4–8 and described in Table 4–9.

Figure 4–8: Interrupt Flag Register

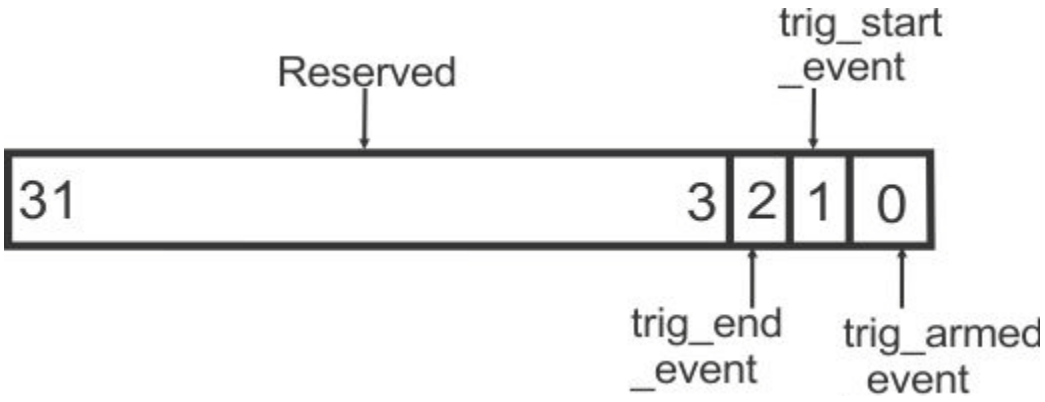


Table 4–9: Interrupt Flag Register (Base Address + 0x1C)

Bits	Field Name	Default Value	Access Type	Description
31:3	Reserved	N/A	N/A	**Reserved**
2	trig_end _event	0	R/CLR	**Trigger End Event Interrupt:** This bit reports the current status of the trigger end event interrupt flag. The trigger end event source indicates that the trigger event has ended. **Read:** 0 = No interrupt 1 = Interrupt latched **Clear:** 1 = Clear latch

Table 4–9: Interrupt Flag Register (Base Address + 0x1C) (Continued)				
Bits	**Field Name**	**Default Value**	**Access Type**	**Description**
1	trig_start _event	0	R/CLR	**Trigger Start Event Interrupt:** This bit reports the current status of the trigger start event interrupt flag. The trigger start event source indicates that a trigger event has started. **Read:** 0 = No interrupt 1 = Interrupt latched **Clear:** 1 = Clear latch
0	trig_armed _event	0	R/CLR	**Trigger Armed Event Interrupt:** This bit reports the current status of the trigger armed event interrupt flag. The trigger armed event interrupt indicates that a trigger condition has been set and generation of the waveform output will begin when the trigger condition has been met. **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 Chirp Generator Core.

5.1 General Design Guidelines

The Pentek Chirp Generator Core provides the required logic to generate a user−configurable chirp waveform which is output as a PDTI−style AXI4−Stream data stream.

5.2 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 both the slave and master AXI4−stream interfaces, as well as the chirp generator.

5.3 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.

AXI4−Stream Reset: **s_axis_aresetn**.

This is an active−low synchronous reset associated with the **s_axis_aclk**. When asserted, both the slave and master AXI4−Stream interfaces are reset, as well as the chirp generator logic.

5.4 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 to create an interrupt.

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.

5.5 Interface Operation

Control/Status Register Interface: This is the control 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. For more details about this interface refer to Section 3.1.

AXI4–Stream Timing Events: This AXI4–Stream Slave Interface provides timestamp, gate, sync and pps data to the core. It is associated with the s_axis_aclk. For more details about this interface refer to Section 3.2.

AXI4–Stream Output: This AXI4–Stream Master Interface provides the output wave–form generated by the core. It is also associated with the s_axis_aclk. For more details about this interface refer to Section 3.2.

5.6 Programming Sequence

This section briefly describes the programming sequence for the AXI4−Stream Chirp Generator Core.

1) Clear the interrupt flags.

2) Set the control register, the frequency value register, the phase offset register, the frequency ramp rate register and the pulse width register to their respective desired values.

3) Enable interrupts as required.

4) Arm the trigger.

5) Observe the output waveform when the trigger condition is met by the incoming timing data.

5.7 Timing Diagrams

The timing diagram for the AXI4−Stream Chirp Generator Core, shown in Figure 6−3, 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 6.5.

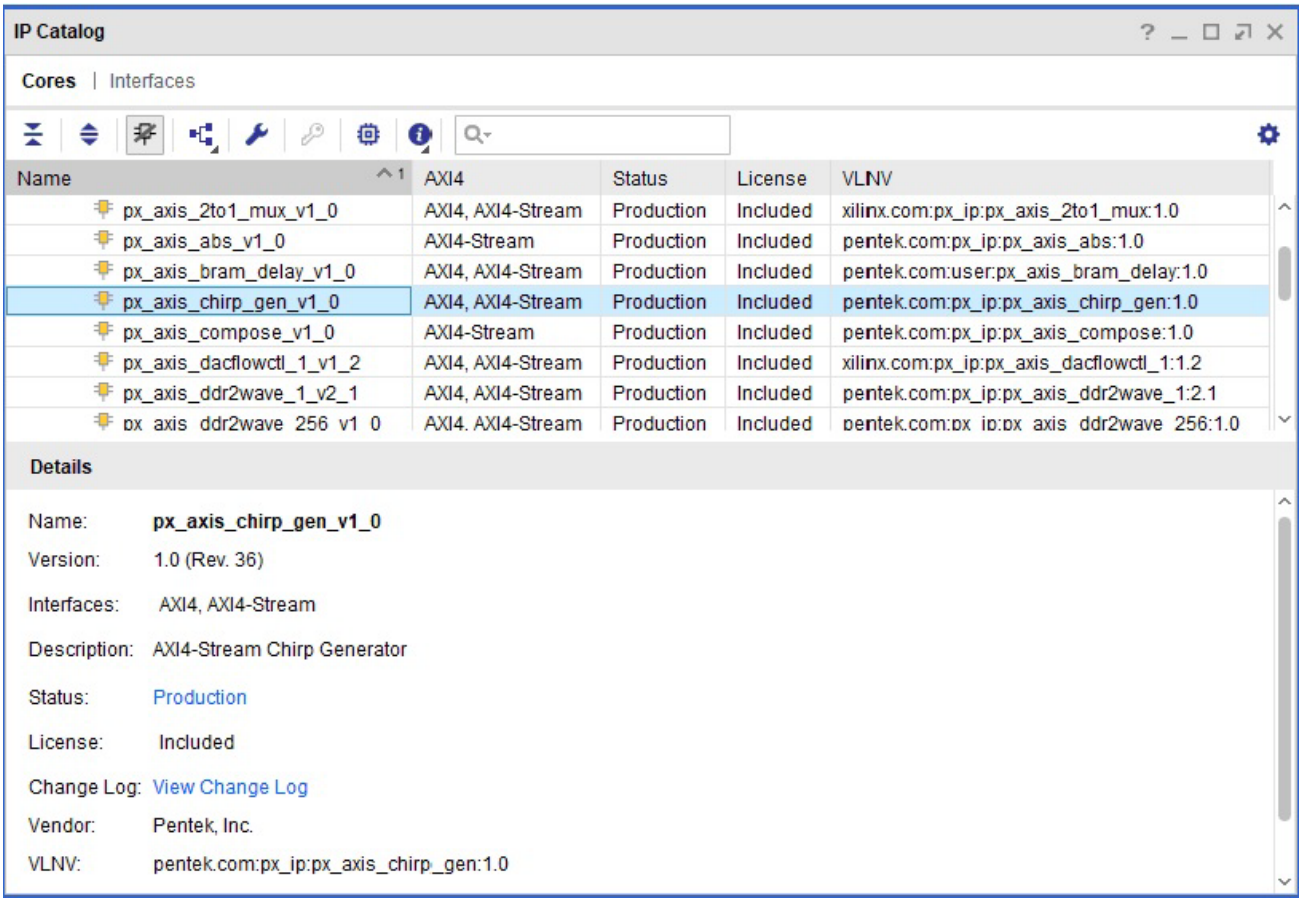
This page is intentionally blank

Chapter 6: Design Flow Steps

6.1 Pentek IP Catalog

This chapter describes customization and generation of the Pentek AXI4–Stream Chirp Generator 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_chirp_gen_v1_0** as shown in Figure 6–1.

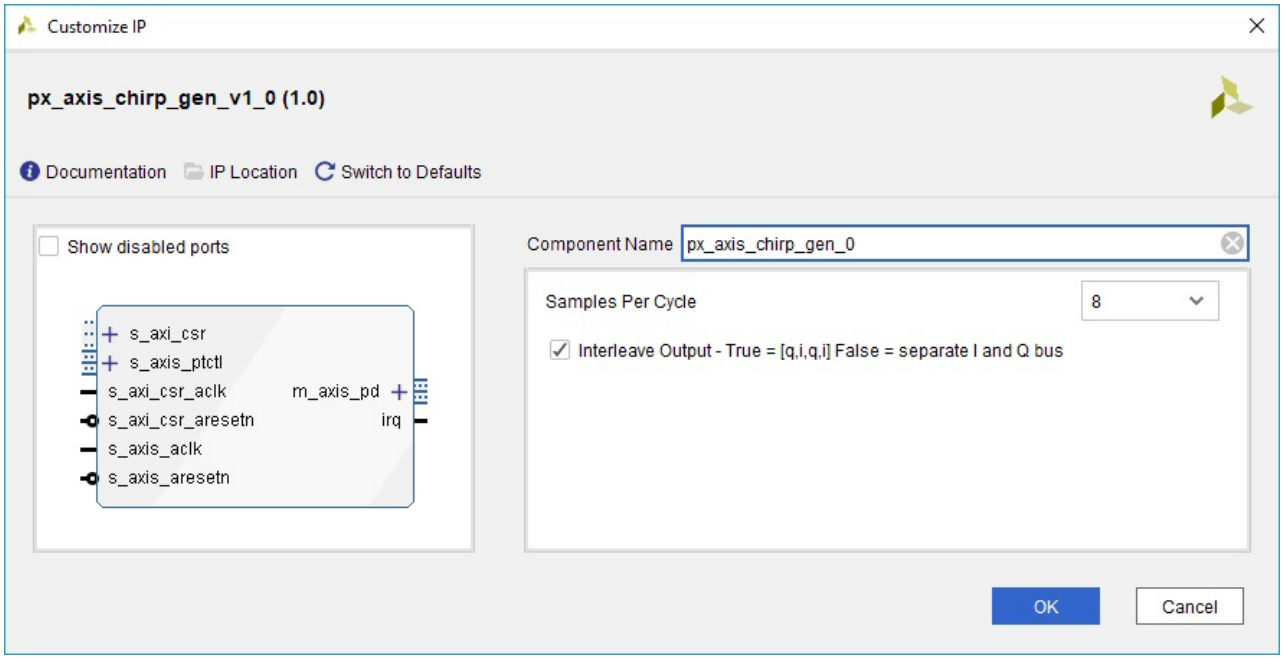
Figure 6–1: AXI4–Stream Chirp Generator Core in Pentek IP Catalog



6.1 Pentek IP Catalog (continued)

When you select the **px_axis_chirp_gen_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 Chirp Generator Core Symbol



6.2 User Parameters

For a detailed explanation of the user parameters, refer to Section 2.5.

6.3 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*.

6.4 Constraining the Core

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

Required Constraints

The XDC constraints are not provided with this core. The necessary constraints can be applied in the top level module of the user design.

Device, Package, and Speed Grade Selections

This IP works for the Kintex Ultrascale FPGAs.

Clock Frequencies

The CSR clock (**`s_axi_csr_aclk`**) has a maximum operating frequency of 250 MHz and the AXI4–Stream clock (**`s_axis_aclk`**) has maximum frequency of 500 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.

6.5 Simulation

The Pentek Chirp Generator Core has a test bench which generates output waveforms using the Vivado VSim environment. The test bench is designed to run with an AXI4–Lite CSR clock (**s_axis_csr_aclk**) frequency of 250 MHz and an AXI4–Stream clock (**s_axis_aclk**) frequency of 500 MHz. The test bench provides parameter values through the test_parameters.txt file, which is located in the top–level of the project directory. The contents of the **test_parameters.txt** file along with descriptions of the parameters are provided in Table 6–1.

Table 6–1: Interrupt Flag Register (Base Address + 0x1C)

Parameter	Type	Value	Description
LOAD_MODE	std_logic _vector	010	**Load Mode:** This parameter indicates the ramp counter and DDS offset load mode based on the source of the PPS signal. 000 – Always on ARM 001 – Aux pulse rising edge 010 – PPS rising edge 011 – PPS falling edge 100 – SYNC rising edge 101 – SYNC falling edge 110 – Gate rising edge 111 – Gate falling edge
STAY_ARMED	Boolean	False	**Stay Armed:** When True this parameter prevents the trigger from being disarmed.
SELECT_SINE	Boolean	False	**Select Sine:** When True, the counter output represents a 64–bit counter output with no pps count. When False, the counter output represents a 32–bit sample count and a 32–bit pps count.
SINE_FREQ _VALUE	Real	25.0	**Sine Frequency Value:** This parameter defines the frequency of the output sine wave in MHz. **NOTE:** This value MUST have a decimal point.
SAMPLE_FREQ	Real	200.0	**Sample Frequency Value:** This is the sample frequency value in MHz. **NOTE:** This value MUST have a decimal point.
TICKS_PER _CYCLE	Integer	1	**Ticks per Cycle:** Clock cycles per **tvalid**

6.5 Simulation (continued)

In addition, the following parameters are set in the test bench:

1) **`interleave_output`** is set to default (TRUE)

2) Registers are initially set as follows:

   a) Control register = 0x80000082: Gain=Unity, **`up_mode`** and **`arm_load`** are enabled

   b) Frequency value = 0x00000000

   c) Frequency ramp rate = 0x00000001

   d) Pulse width = 0x00000200

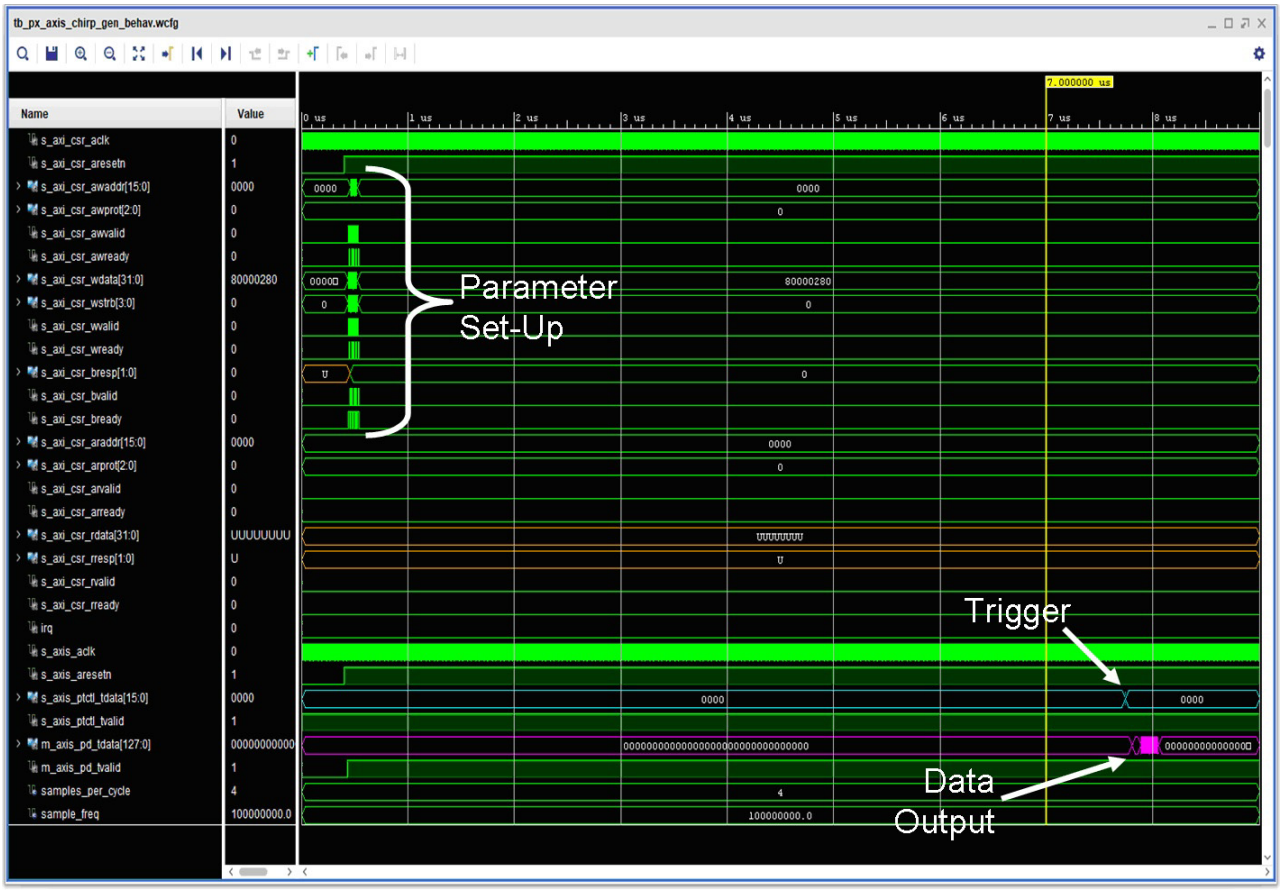
3) No interrupts are enabled

Once the reset is released, the testbench writes 0x80000280 (unity gain, with **`up_mode`** and **`calc_start`** are enabled) to the control register to enable waveform generation upon a trigger event. Several µseconds later a trigger pattern is placed on the **`s_axis_ptctl_tdata`** bus. Shortly thereafter the waveform data begins to appear on the **`m_axis_pd_tdata`** bus.

When run, the simulation produces the results shown in Figure 6–3.

6.5 Simulation (continued)

Figure 6-3: AXI4-Stream Chirp Generator Core Test Bench Simulation Output



6.6 Synthesis and Implementation

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