IP CORE MANUAL

NAVIGATOR

FPGA Design Kit

Test Signal Generator IP

px_test_sig_gen

PENTEK

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

Manual Part Number: 807.48328

Rev: 1.0 - December 09, 2016

Manual Revision History

Date	Version	Comments
12/09/16	1.0	Initial Release

Legal Notices

The information disclosed to you hereunder (the "Materials") is provided solely for the selection and use of Pentek products. To the maximum extent permitted by applicable law: (1) Materials are made available "AS IS" and with all faults, Pentek hereby DISCLAIMS ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE; and (2) Pentek shall not be liable (whether in contract or tort, including negligence, or under any other theory of liability) for any loss or damage of any kind or nature related to, arising under, or in conjunction with, the Materials (including your use of Materials), including for any direct, indirect, special, incidental, or consequential loss or damage (including loss of data, profits, goodwill, or any type of loss or damage suffered as a result of any action brought by a third party) even if such damage and loss was reasonably foreseeable or Pentek had been advised of the possibility of the same. Pentek assumes no obligation to correct any error contained in the Materials or to notify you of updates to the Materials or to product specifications. You may not reproduce, modify, distribute, or publicly display the materials without prior written consent. Certain products are subject to the terms and conditions of Pentek's limited warranty, please refer to Pentek's Ordering and Warranty information which can be viewed at http://www.pentek.com/contact/customerinfo.cfm; IP cores may be subject to warranty and support terms contained in a license issued to you by Pentek. Pentek products are not designed or intended to be fail-safe or for use in any application requiring fail-safe performance; you assume sole risk and liability for the use of Pentek products in such critical applications.

Copyright

Trademarks

Pentek, Jade, and Navigator are trademarks or registered trademarks of Pentek, Inc.

ARM and AMBA are registered trademarks of ARM Limited. PCI, PCI Express, PCIe, and PCI-SIG are trademarks or registered trademarks of PCI-SIG. Xilinx, Kintex UltraScale, Vivado, and Platform Cable USB are registered trademarks of Xilinx Inc., of San Jose, CA.

Printed in the United States of America.

Table of Contents

Page

IP Facts

Chapter 1: Overview

Chapter 2: General Product Specifications

Chapter 3: Port Descriptions

Table of Contents

Page

Chapter 4: Register Space

Chapter 5: Designing with the Core

Chapter 6: Design Flow Steps

IP Facts

Description

Pentek's Navigator™ Test Signal Generator Core generates digitized ramp or programmable frequency sine wave signals for use as test signals.

This core complies with the ARM® AMBA® AXI4 Specification and also provides a control/status register interface. This user manual defines the hardware interface, software interface, and parameterization options for the Test Signal Generator Core.

Features

- Generates programmable frequency sine wave or ramp signals
- Register access through AXI4-Lite interface
- Software programmable number of samples per clock cycle in the output data
- Includes a state machine to control the load on the Ramp Counter and Direct Digital Synthesizer (DDS) Compiler
- Provides register access to control the load mode and desired output selection (ramp/sine wave)

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 2016.3 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

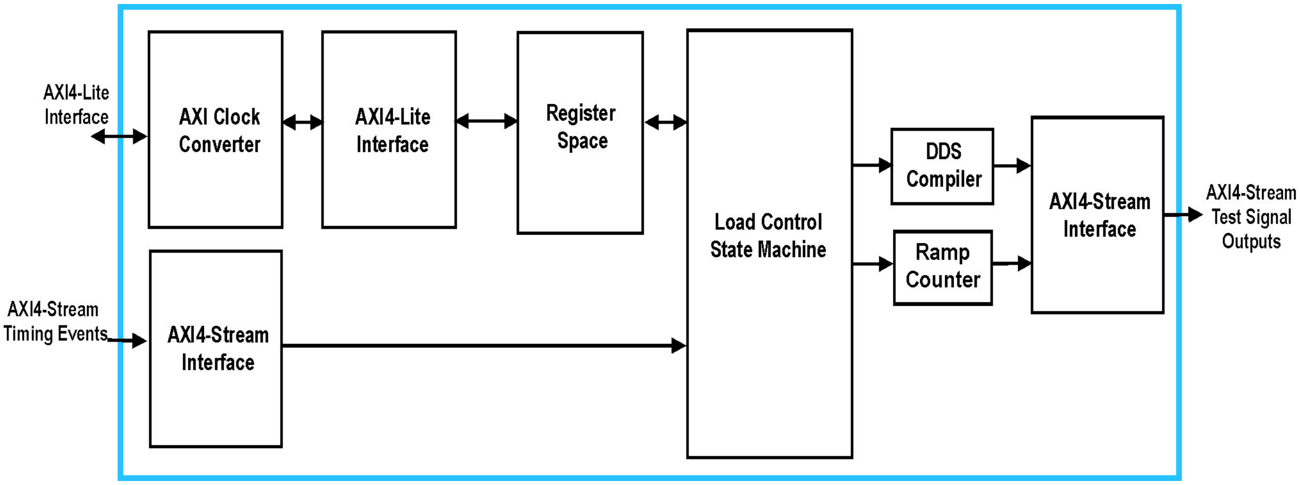
The Test Signal Generator Core accepts input Timing Event Streams (Sample Clock, Reset and Timing Signals) through an AXI4-Stream Slave Interface and delivers output Test Signal Streams through an AXI4-Stream Master Interface. Key components of the Test Signal Generator Core are:

- **AXI Clock Converter Core** - This is connected to the AXI4-Lite Interface, as shown in Figure 1-1, in order to operate the Register Space in the Sample Clock domain.
- **DDS Compiler** - Generates the desired sine wave output.
- **Ramp Counter** - Generates the desired ramp output.
- **Counter Load Control State Machine** - Controls the offset values based on the load mode selected using the Mode Control Register (see Section 4.1).

The output Test Signal data stream is generated based on the Mode Control Register setting of the output select bit (**output_sel**). For more details on the Test Signal output from the AXI4-Stream Master Interface, refer to Section 3.2.

Figure 1-1 is a top-level block diagram of the Pentek Test Signal Generator Core. The modules within the block diagram are explained in the later sections of this manual.

Figure 1-1: Test Signal Generator Core Block Diagram



1.1 Functional Description (continued)

- ❑ **AXI Clock Converter Core:** The AXI Clock Converter Core is included in the Xilinx AXI Interconnect Core and is used to connect one AXI memory-mapped slave to an AXI memory-mapped master which is operating in a different clock domain. In the Test Signal Generator Core, the AXI Clock Converter is used to operate the Register Space in the Sample Clock domain.

- ❑ **AXI4-Stream Interface:** The Test Signal Generator Core has two AXI4-Stream Interfaces. At the input, an AXI4-Stream Slave Interface is used to receive input timing event data streams and at the output an AXI4-Stream Master Interface is used to transfer test signal data streams through the output ports. For more details about the AXI4-Stream Interfaces please 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 additional details about the AXI4-Lite Interface, refer to Section 3.1 AXI4-Lite Core Interfaces.

- ❑ **Register Space:** This module contains control and status registers, including Interrupt Enable, Interrupt Status and Interrupt Flag registers. Registers are accessed through the AXI4-Lite interface.

- ❑ **Counter Load Control State Machine:** This state machine is used to control the offset load of the Ramp Counter and DDS Compiler and has three states:

  - **Reset** - The Reset state resets the state machine based on the input reset signal (**s_axis_aresetn**) from the input AXI4-Stream Slave Interface.

  - **Wait for Arm** - When the state machine is in the Wait for Arm state, the Test Signal Generator Core waits for the arm load signal, from the Mode Control Register, to go High.

  - **Armed** - Once in the Armed State, the core waits for an edge event on the PPS signal and loads the counter and DDS with the required offset values when the event occurs. The state machine then goes back to the **Wait for Arm** state while the test signal data output is being generated. To reset the load offset on the DDS and the Ramp counter with a new value, the **arm load bit** must be set to 1.

- ❑ **Direct Digital Synthesizer:** The Xilinx Direct Digital Synthesizer (DDS) Compiler core is used to generate the required output test signal sinusoidal waveforms. For more details about the DDS Compiler core please refer to the *Xilinx Direct Digital Synthesizer Compiler Core: Product Guide*.

- ❑ **Ramp Counter:** The Ramp Counter provides incrementing output test signal ramp data stream.

1.2 Applications

The Test Signal Generator Core can be used for generating AXI4 Test Signal Streams and can be incorporated into any Kintex Ultrascale FPGAs.

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

4) *Xilinx Direct Digital Synthesizer Compiler Core: Product Guide*

This page is intentionally blank

Chapter 2: General Product Specifications

2.1 Standards

The Test Signal Generator Core has bus interfaces that comply with the *ARM AMBA AXI4-Lite Protocol Specification* and the *AMBA AXI4-Stream Protocol Specification*.

2.2 Performance

The performance of the Test Signal 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

The Test Signal Generator core has two incoming clock signals. The input Sample clock has a maximum frequency of 500 MHz while the clock across the AXI4-Lite interface has a maximum 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 Test Signal 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	686
Flip-Flops	1465
Memory LUTs	78
DSP	2

NOTE: Actual utilization may vary based on the user design in which the Test Signal Generator Core is incorporated.

2.4 Limitations and Unsupported Features

This section is not applicable to this IP core.

2.5 Generic Parameters

The generic parameters of the Test Signal 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	**Sample per Clock Cycle:** This parameter indicates the number of samples per clock cycle in the output data stream. It can take the values 1, 2, 4, and 8.

Chapter 3: Port Descriptions

This chapter provides details about the port descriptions for the following interface types:

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

3.1 AXI4-Lite Core Interfaces

The Test Signal Generator Core uses the Control/Status Register (CSR) interface to control, and receive status 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 and status registers in the Test Signal Generator Core. Table 3-1 defines the ports in the CSR interface. See Chapter 4 for a Control/Status Register memory map and bit definitions. See the *AMBA AXI4-Lite Specification* for more details on operation of the AXI4-Lite interfaces.

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 signal will reset all control registers to their initial states.
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 Test Signal generator core.
s_axi_csr_awprot	Input	3	**Protection:** The Test Signal 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 Test Signal 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**.

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 Test Signal 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.
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 Test Signal 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 Test Signal 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 Test Signal 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.

Table 3-1: Control/Status Register (CSR) Interface Port Descriptions (Continued)			
Port	**Direction**	**Width**	**Description**
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 Test Signal generator core.
s_axi_csr_arprot	Input	3	**Protection:** These bits are ignored by the Test Signal generator 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 the **s_axi_csr_araddr**. The Test Signal Generator 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 Test Signal 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 Test Signal 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 Test Signal 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 Test Signal Generator core has the following AXI4-Stream Interfaces, used to receive and transfer data streams.

- Timing Events (PTCTL) Interface: The interface through which Timing Events are received.
- Test Signal Data Stream (PD_TESTSIG) Interface: The interface through which Test Signal data streams are transferred through the output ports.

3.2.1 Timing Events (PTCTL) Interface

Table 3-2 defines the ports in the Timing Events Interface. This interface is an AXI4-Stream Slave Interface. See the *AMBA AXI4-Stream Specification* for more details on the operation of the AXI4-Stream Interface.

Table 3-2: Timing Events Interface Port Descriptions

Port	Direction	Width	Description
s_axis_aclk	Input	1	**Sample Clock**
s_axis_aresetn	Input		**Reset:** Active Low.
s_axis_ptctl_tdata	Input	depends on the generic parameter **samples_per_cycle**	**Input Data:** This is timing event data which indicates the Gate/ Sync/ PPS signal positions. tdata[n-1:0] - Gate Positions tdata[2n-1:n] - Sync Positions tdata[3n-1:2n] - PPS Positions
s_axis_ptctl_tvalid	Input	1	**Input Data Valid:** Asserted when data is valid on **s_axis_ptctl_tdata**.

3.2 AXI4-Stream Core Interfaces (continued)

3.2.2 Test Signal Data Stream (PD_TESTSIG) Interface

Table 3-3 defines the ports in the Test Signal Data Stream Interface. This interface is an AXI4-Stream Master Interface through which the generated test signals are transferred over the output ports. See the *AMBA AXI4-Stream Specification* for more details on the operation of the AXI4-Stream Interface.

Table 3-3: Test Signal Data Stream Interface Port Descriptions

Port	Direction	Width	Description
m_axis_pd_testsig_tdata	Output	depends on the generic parameter **samples_per_cycle**	**Output Test signal Data Stream:** This is the Test Signal output of the Test Signal Generator Core.
m_axis_pd_testsig_tvalid		1	**Output Test Signal Data Valid:** Asserted when data is valid on **m_axis_pd_testsig_tdata.**

This page is intentionally blank

Chapter 4: Register Space

This chapter provides the memory map and register descriptions for the register space of the Test Signal Generator Core. The memory map is provided in Table 4-1.

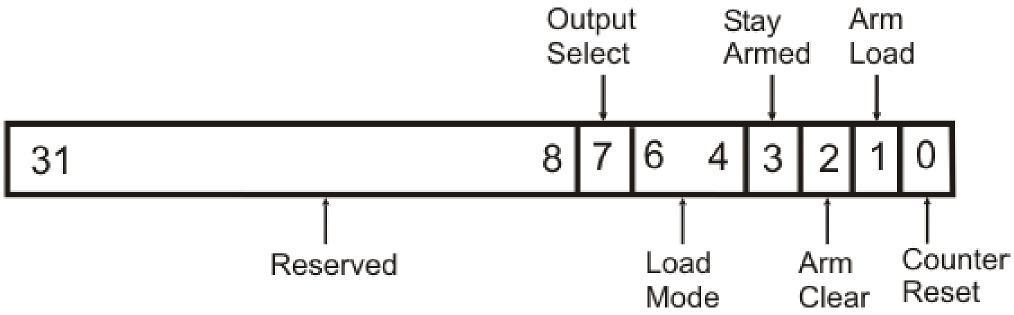
Table 4-1: Register Space Memory Map

Register Name	Address (Base Address +)	Access	Description
Mode Control	0x00	R/W	Controls the mode of operation of the Test Signal Generator Core.
Sine Wave Frequency Value	0x04		Controls frequency of the output sine wave.
Reserved	0x08		Reserved
	0x0C	R	
	0x10		
Interrupt Enable Register	0x14	R/W	Interrupt enable bits
Interrupt Status Register	0x18	R	Interrupt source status bits
Interrupt Flag Register	0x1C	R/Clr	Interrupt flag bits

4.1 Mode Control Register

This control register is used to control the Counter reset, ARM load and Clear, Stay Armed, Load Mode, and Output Select bits of the Test Signal Generator 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
31:8	Reserved	N/A	N/A	**Reserved**
7	output_sel	0	R/W	**Output Select:** This bit is used to select the desired output from the Test Signal Generator Core. 0 = Ramp output 1 = Sine wave output
6:4	load_mode	000	R/W	**Load Mode:** These bits indicate 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
3	stay_armed	0	R/W	**Stay Armed:** When set to ‘1’, this bit keeps the Counter Load Control State Machine in the Armed state with the **load_armed** status signal High. 0 = State machine goes to Wait for Arm state 1 = State machine remains in Armed state

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

Bits	Field Name	Default Value	Access Type	Description
2	arm_clear	0	R/W	**ARM Clear:** This is used to clear the offset load on the ramp counter and the DDS. 0 = Remain unchanged 1 = Clear
1	arm_load	0	R/W	**ARM Load:** When this bit is toggled ‘1’ then ‘0’, it changes the state of the load control state machine to the Armed state, and enables an offset load on the ramp counter and the DDS when the arm clear bit of this register is ‘0’.
0	cntr_rst	0	R/W	**Counter Reset:** This bit is used to reset the ramp counter and the DDS compiler core. 0 = Run 1 = Reset

4.2 Sine Wave Frequency Value Register

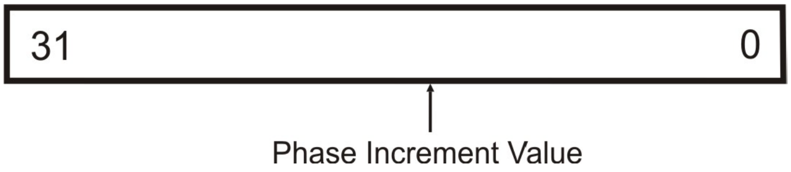
This register controls the phase increment of the sine signal generated by the Xilinx DDS Core. This value is used to determine the frequency of the sine output. The output frequency is given by

$F_{(out)}$ = ($F_{(clk)}$ * Phase increment value) / (2^32)

Here $F_{(clk)}$ = Sample clock frequency

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

Figure 4-2: Sine Wave Frequency Register



.

Table 4-3: Sine Wave Frequency Value Register (Base Address + 0x04)

Bits	Field Name	Default Value	Access Type	Description
31:0	phase_inc _val	0x00000000	R/W	**Phase Increment Value:** This is the phase increment of the Xilinx DDS core which inturn determines the sine wave output frequency.

4.3 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 4.4). The Interrupt Enable Register is illustrated in Figure 4-3 and described in Table 4-4.

Figure 4-3: Interrupt Enable Register

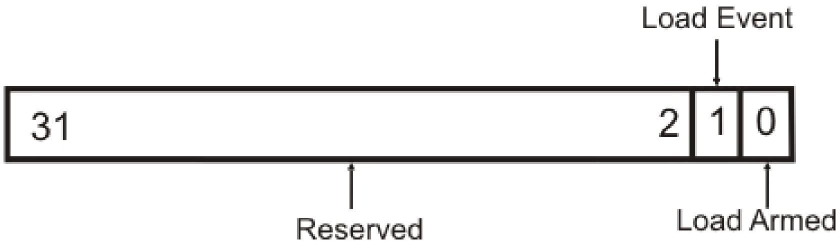


Table 4-4: Interrupt Enable Register (Base Address + 0x14)

Bits	Field Name	Default Value	Access Type	Description
31:2	Reserved	N/A	N/A	**Reserved**
1	load_event	0	R/W	**Load Event:** This bit enables/disables the load event interrupt source. The load event interrupt source indicates that a load event has occurred, and the ramp counter and DDS compiler are loaded with offset values to generate the test signal output. The offset values are generated within the core based on the input timing event AXI streams and the value in the sinewave frequency value register. 0 = Disable interrupt 1 = Enable interrupt
0	load_armed			**Load Armed:** This bit enables/disables the load armed interrupt source. The load event interrupt source indicates that the Counter Load Control state machine is in the Armed state. 0 = Disable interrupt 1 = Enable interrupt

4.4 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 4-3 and described in Table 4-4.

Figure 4-4: Interrupt Status Register

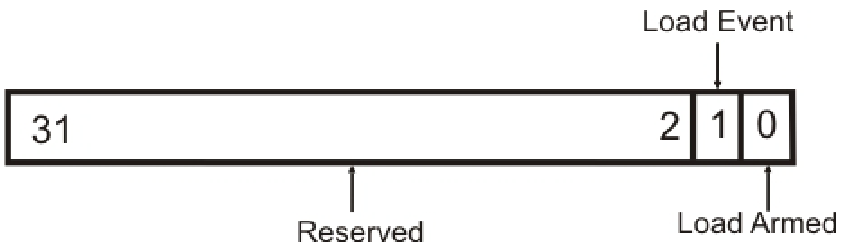


Table 4-5: Interrupt Status Register (Base Address + 0x18)

Bits	Field Name	Default Value	Access Type	Description
31:2	Reserved	N/A	N/A	**Reserved**
1	load_event	0	R	**Load Event:** This bit indicates the status of the load event interrupt source. The load event interrupt source indicates that a load event has occurred, and the ramp counter and DDS compiler are loaded with offset values to generate the test signal output. 0 = No interrupt 1 = Interrupt condition asserted
0	load_armed			**Load Armed:** This bit indicates the status of the load armed interrupt source. The load event interrupt source indicates that the Counter Load Control state machine is in the Armed state. 0 = No interrupt 1 = Interrupt condition asserted

4.5 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 enable register. The Interrupt Status Register is illustrated in Figure 4-3 and described in Table 4-4.

Figure 4-5: Interrupt Flag Register

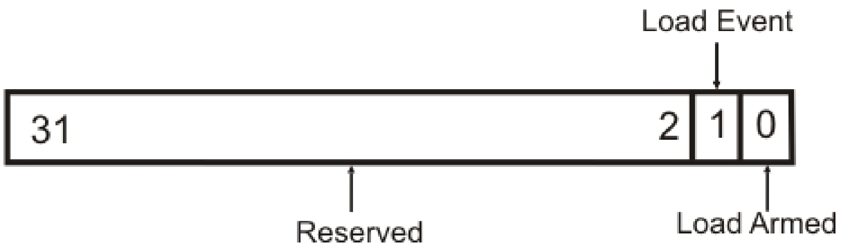


Table 4-6: Interrupt Flag Register (Base Address + 0x1C)				
Bits	**Field Name**	**Default Value**	**Access Type**	**Description**
31:2	Reserved	N/A	N/A	**Reserved**
1	load_event	0	R/Clr	**Load Event:** This bit indicates the load event interrupt flag. The load event interrupt source indicates that a load event has occurred, and the ramp counter and DDS compiler are loaded with offset values to generate the test signal output. **Read:** 0 = No interrupt 1 = Interrupt latched **Clear:** 1 = Clear latch
0	load_armed			**Load Armed:** This bit indicates the load armed interrupt flag. The load event interrupt source indicates that the Counter Load Control state machine is in the Armed state. **Read:** 0 = No interrupt 1 = Interrupt latch **Clear:** 1 = Clear latch

This page is intentionally blank

Chapter 5: Designing with the Core

This chapter includes guidelines and additional information to facilitate designing with the Test Signal Generator Core.

5.1 General Design Guidelines

The Test Signal Generator Core provides the required logic to generate Test Signal output data. This IP core supports AXI4-Lite and AXI4-Stream user interfaces. The user can generate the required output from the core based on the setting of the generic parameter, as described in Section 2.5, and the Mode Control Register.

5.2 Clocking

Sample Clock: **s_axis_aclk**

This clock is used to clock all ports of the core.

CSR Clock: **s_axi_csr_aclk**

This clock is the input AXI4-Lite interface clock to the core which is converted using the AXI Clock converter core to operate the other modules within the Test Signal Generator Core in the Sample Clock domain.

5.3 Resets

Main reset: **s_axis_aresetn**

This is an active low synchronous reset associated with **s_axis_aclk**. When asserted, the state machine in the Test Signal Generator core is reset.

CSR Reset: **s_axi_csr_aresetn**

This is an active low reset synchronous with **s_axi_csr_clk**. When asserted, the control/status registers and the interrupt registers are reset.

5.4 Interrupts

This core has an edge-type (rising edge-triggered) interrupt output. It is synchronous with the **s_axis_aclk**. On the rising edge of any interrupt signal, a one clock cycle wide pulse is output from the core on it's **irq** output. Each interrupt event is stored in two registers, accessible on the **s_axi_csr** bus.

5.4 Interrupts (continued)

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

CSR Interface: This is the Control/Status Register Interface and is associated with **s_axis_aclk**. It is a standard AXI4-Lite Slave interface. See Chapter 4 for the control register memory map, which provides more details on the registers that can be accessed through this interface.

Timing Events (PTCTL) Interface: This is the interface through which Timing Events are received. It is an AXI4-Stream Slave interface and is associated with **s_axis_aclk**. For more details about this interface, refer to Section 3.2.1.

Test Signal Data Stream (PD_TESTSIG) Interface: This is the interface through which Test Signal outputs are transferred through the output ports. It is an AXI4-Stream Master interface. For more details about this interface, refer to Section 3.2.2.

5.6 Programming Sequence

This section briefly describes the programming sequence of registers in the Test Signal Generator Core.

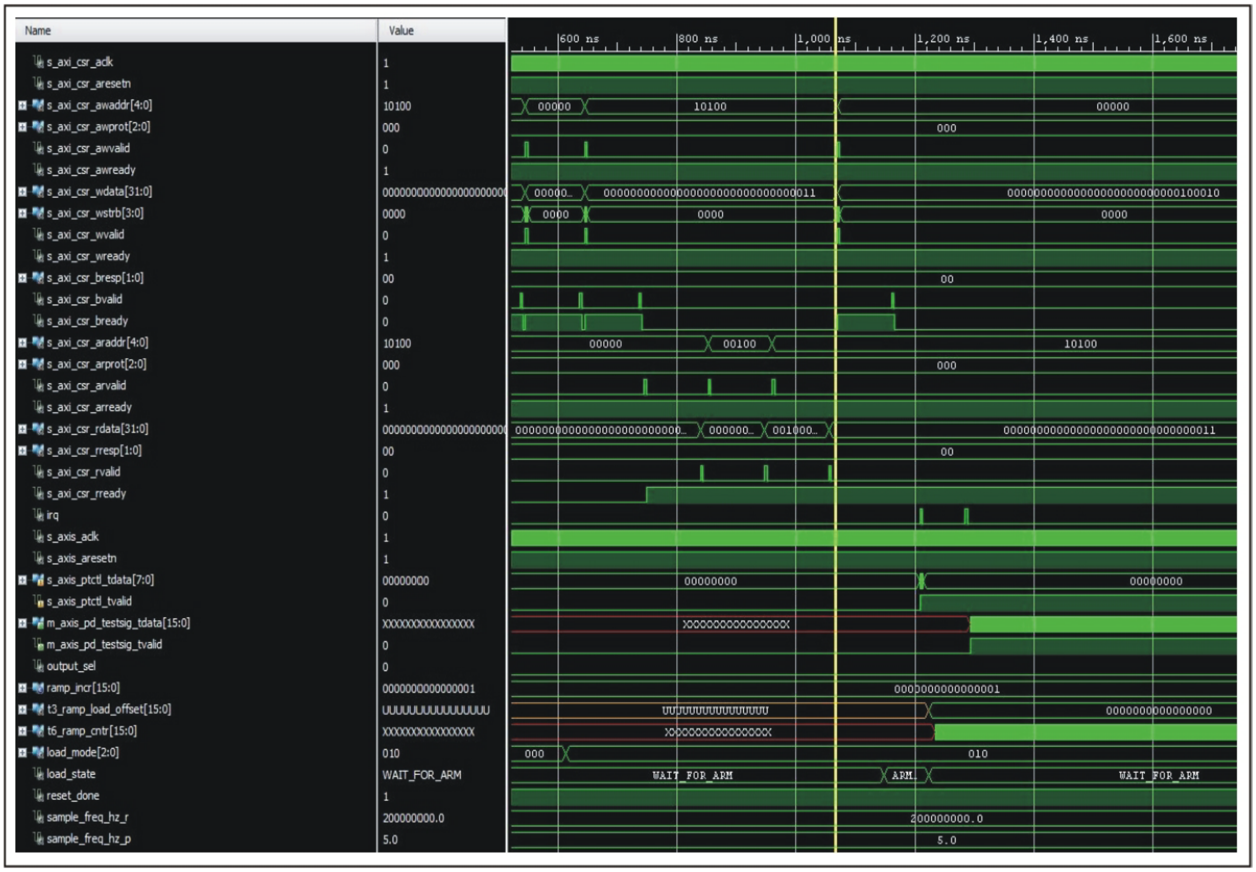
1) Ensure that the interrupt flag register is cleared.
2) Enable the interrupt enable bits based on the user design requirement.
3) Assign the desired value to the generic parameter, Samples per Clock Cycle (see Section 2.5).
4) Set the Mode Control Register with the required values.
5) Observe the outputs across the outputs ports.
6) When done check the interrupt flag register and clear the interrupts.

5.7 Timing Diagrams

The timing diagrams for the Test Signal Generator Core are 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.

The timing diagrams depict the functionality of the core for Output Select bit of the Mode Control Register set to 0 and 1 i.e., for an output ramp signal and output sine wave signal from the Test Signal Generator Core. The timing diagrams for the different outputs can be observed in Figures 5-1 and 5-2.

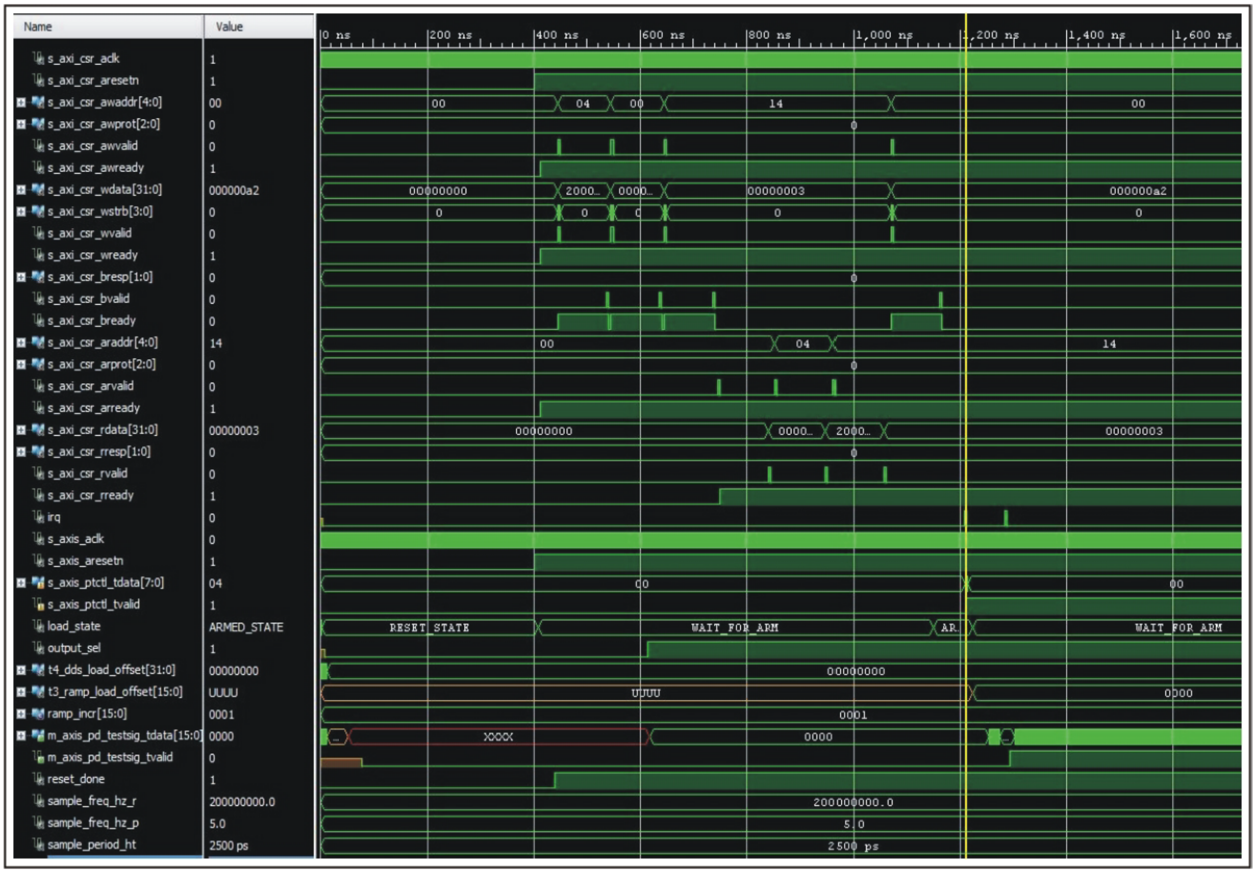
Figure 5-1: Test Signal Generator IP Core Timing Diagram - Ramp Output



Name	Value
s_axi_csr_aclk	1
s_axi_csr_aresetn	1
s_axi_csr_awaddr[4:0]	10100
s_axi_csr_awprot[2:0]	000
s_axi_csr_awvalid	0
s_axi_csr_awready	1
s_axi_csr_wdata[31:0]	0000000000000000000000
s_axi_csr_wstrb[3:0]	0000
s_axi_csr_wvalid	0
s_axi_csr_wready	1
s_axi_csr_bresp[1:0]	00
s_axi_csr_bvalid	0
s_axi_csr_bready	0
s_axi_csr_araddr[4:0]	10100
s_axi_csr_arprot[2:0]	000
s_axi_csr_arvalid	0
s_axi_csr_arready	1
s_axi_csr_rdata[31:0]	0000000000000000000000
s_axi_csr_rresp[1:0]	00
s_axi_csr_rvalid	0
s_axi_csr_rready	1
irq	0
s_axis_aclk	1
s_axis_aresetn	1
s_axis_ptctl_tdata[7:0]	00000000
s_axis_ptctl_tvalid	0
m_axis_pd_testsig_tdata[15:0]	XXXXXXXXXXXXXXXX
m_axis_pd_testsig_tvalid	0
output_sel	0
ramp_incr[15:0]	0000000000000001
t3_ramp_load_offset[15:0]	UUUUUUUUUUUUUUUU
t6_ramp_cntr[15:0]	XXXXXXXXXXXXXXXX
load_mode[2:0]	010
load_state	WAIT_FOR_ARM
reset_done	1
sample_freq_hz_r	200000000.0
sample_freq_hz_p	5.0

5.7 Timing Diagrams (continued)

Figure 5-2: Test Signal Generator IP Core Timing Diagram - Sine Wave Output



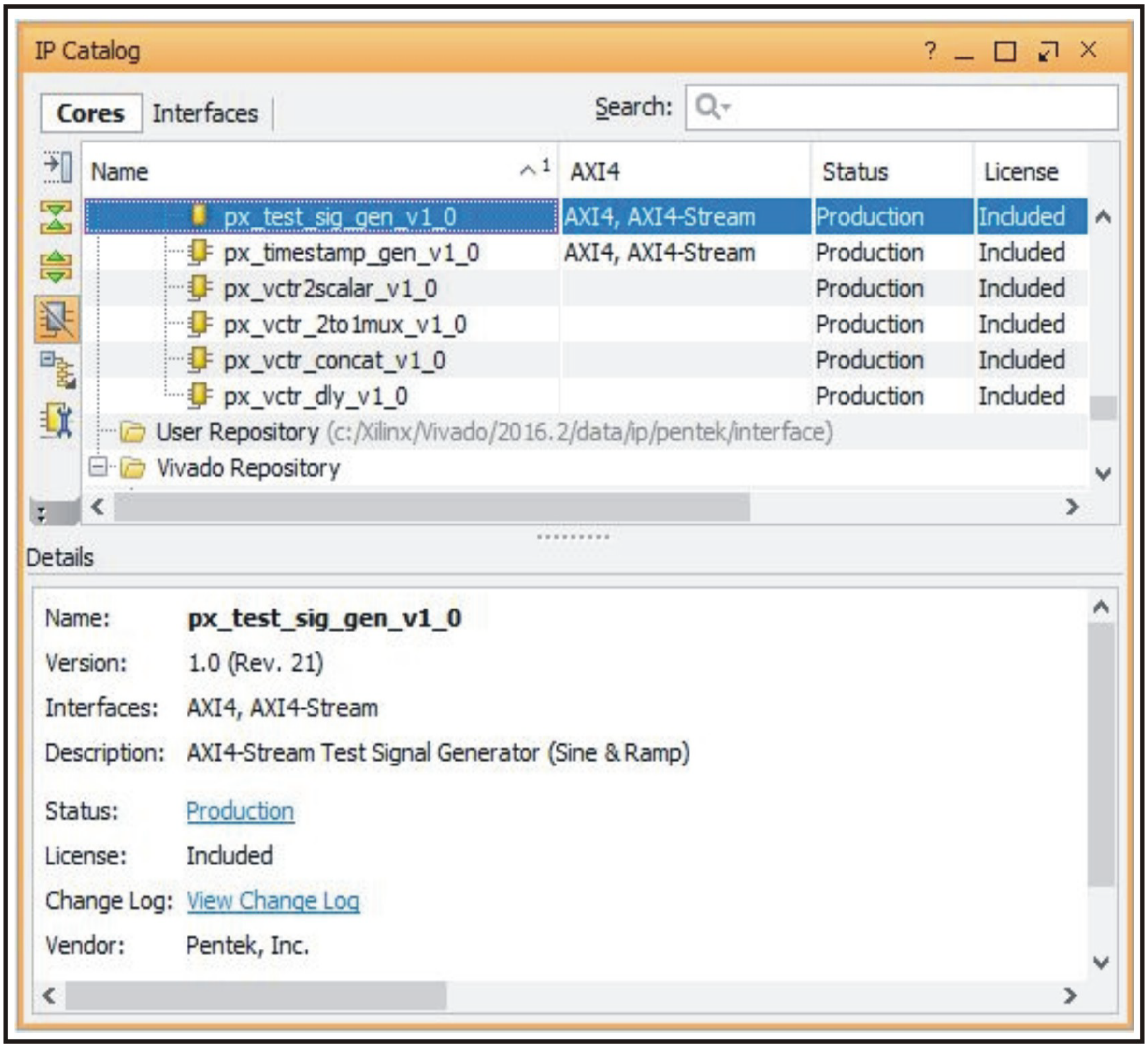
Name	Value
s_axi_csr_aclk	1
s_axi_csr_aresetn	1
s_axi_csr_awaddr[4:0]	00
s_axi_csr_awprot[2:0]	0
s_axi_csr_awvalid	0
s_axi_csr_awready	1
s_axi_csr_wdata[31:0]	000000a2
s_axi_csr_wstrb[3:0]	0
s_axi_csr_wvalid	0
s_axi_csr_wready	1
s_axi_csr_bresp[1:0]	0
s_axi_csr_bvalid	0
s_axi_csr_bready	0
s_axi_csr_araddr[4:0]	14
s_axi_csr_arprot[2:0]	0
s_axi_csr_arvalid	0
s_axi_csr_arready	1
s_axi_csr_rdata[31:0]	00000003
s_axi_csr_rresp[1:0]	0
s_axi_csr_rvalid	0
s_axi_csr_rready	1
irq	0
s_axis_aclk	0
s_axis_aresetn	1
s_axis_ptctl_tdata[7:0]	04
s_axis_ptctl_tvalid	1
load_state	ARMED_STATE
output_sel	1
t4_dds_load_offset[31:0]	00000000
t3_ramp_load_offset[15:0]	UUUU
ramp_incr[15:0]	0001
m_axis_pd_testsig_tdata[15:0]	0000
m_axis_pd_testsig_tvalid	0
reset_done	1
sample_freq_hz_r	200000000.0
sample_freq_hz_p	5.0
sample_period_ht	2500 ps

Chapter 6: Design Flow Steps

6.1 Pentek IP Catalog

This chapter describes customization and generation of the Pentek Test Signal 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_test_sig_gen_v1_0** as shown in Figure 6-1.

Figure 6-1: Test Signal Generator Core in Pentek IP Catalog



Name	AXI4	Status	License
px_test_sig_gen_v1_0	AXI4, AXI4-Stream	Production	Included
px_timestamp_gen_v1_0	AXI4, AXI4-Stream	Production	Included
px_vctr2scalar_v1_0		Production	Included
px_vctr_2to1mux_v1_0		Production	Included
px_vctr_concat_v1_0		Production	Included
px_vctr_dly_v1_0		Production	Included
User Repository (c:/Xilinx/Vivado/2016.2/data/ip/pentek/interface)			
Vivado Repository			

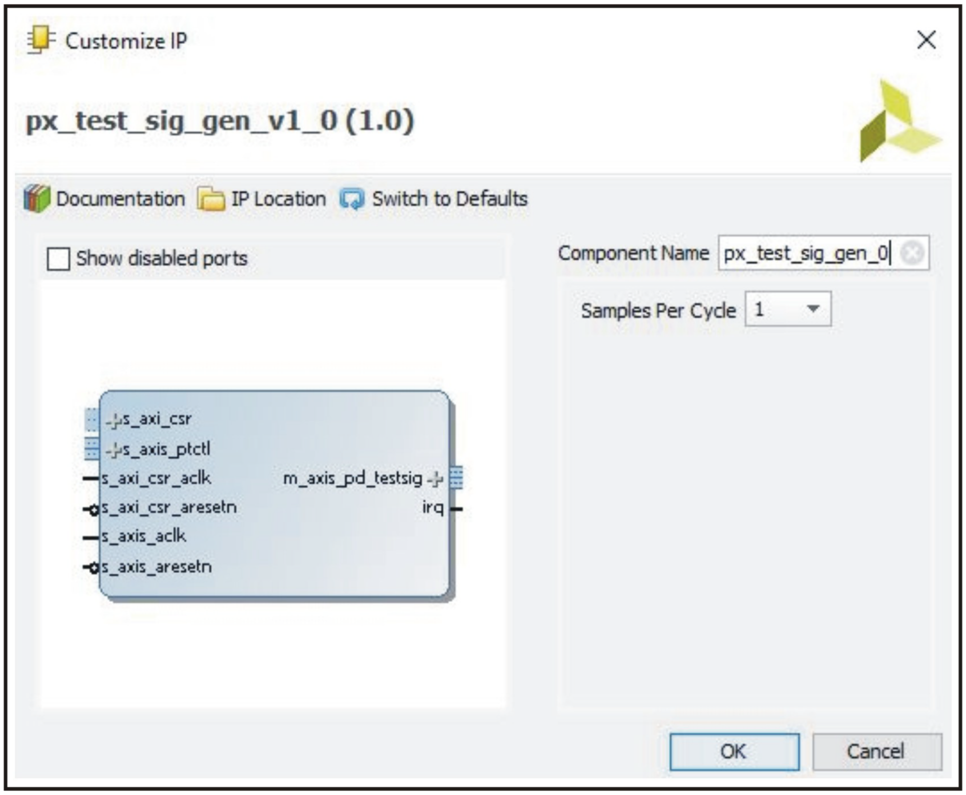
Details

Name: **px_test_sig_gen_v1_0**
Version: 1.0 (Rev. 21)
Interfaces: AXI4, AXI4-Stream
Description: AXI4-Stream Test Signal Generator (Sine & Ramp)
Status: Production
License: Included
Change Log: View Change Log
Vendor: Pentek, Inc.

6.1 Pentek IP Catalog (continued)

When you select the **px_test_sig_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: Test Signal Generator Core IP Symbol



Customize IP

px_test_sig_gen_v1_0 (1.0)

Component Name: px_test_sig_gen_0

Samples Per Cycle: 1

6.2 User Parameters

The user parameters of this IP core are described in Section 2.5 of this user manual.

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 Test Signal Generator Core in Vivado Design Suite.

Required Constraints

The XDC constraints are not provided with the Test Signal Generator Core. Clock 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 clock (**s_axi_csr_aclk**) can take frequencies up to 250 MHz. The sample clock (**s_axis_aclk**) has a 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 Test Signal Generator Core has a test bench which generates the output waveforms using the Vivado VSim environment. The test bench is designed to run at 250 MHz input clock frequency and Sample clock frequency of 200 MHz. The test bench runs for 1 sample per clock cycle and provides parameter values through the **test_parameters.txt** file.

6.5 Simulation (continued)

The contents of the **test_parameters.txt** file along with descriptions of the parameters are discussed in Table 6-1.

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

Parameter	Type	Value	Description
load_mode	std_logic_ vector	0x2	**Counter 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 set to High, this signal keeps the state machine in the Armed state and enables counter loading.
select_sine		True	**Sinewave Select:** This bit is used to select the desired output from the Test Signal Generator core. 0 = Ramp output 1 = Sine wave output
sine_freq_val	Real	25	**Sine Wave Frequency:** This parameter defines the frequency of the sine wave output of this core. It is defined in MHz.
sample_freq		200	**Sample Frequency:** This parameter defines the sample clock frequency and is defined in MHz.
ticks_per_cycle	Integer	1	**Clock cycles per tvalid**

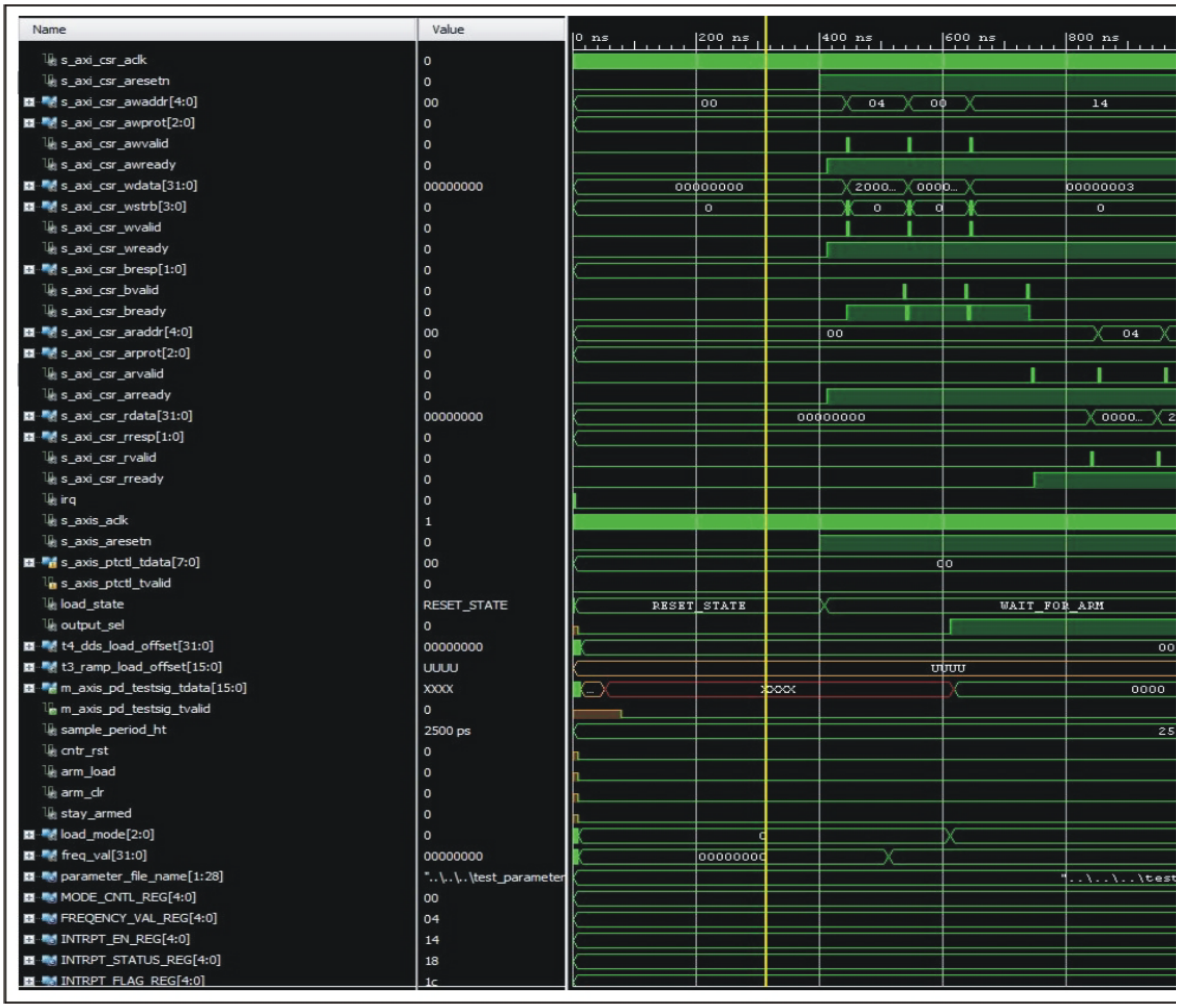
The test bench also assigns the required values to the control registers and Interrupt Enable Register. The test bench runs with the **Output Select** bit in the Mode Control Register set to 1 and at a sine wave frequency of 25 MHz. The **Arm Load** bit of the Mode Control Register is set High for one clock cycle.

During this clock cycle, the **Load Control** state machine moves from the **Wait for Arm** state to the **Armed** state. In this state, it waits for the rising edge of the PPS signal. Once the PPS signal has a rising edge event, the state machine loads the offset value into the **Ramp Counter** and the **DDS Compiler** and then moves to the **Wait for Arm** state.

6.5 Simulation (continued)

The output is generated based on the **Output Select** signal bit set in the Mode Control Register. The programming procedure is the same as described in Section 5.6. When run, the simulation produces the results shown in Figure 6-3.

Figure 6-3: Test Signal Generator Core Test Bench Simulation Output



Name	Value
s_axi_csr_aclk	0
s_axi_csr_aresetn	0
s_axi_csr_awaddr[4:0]	00
s_axi_csr_awprot[2:0]	0
s_axi_csr_awvalid	0
s_axi_csr_awready	0
s_axi_csr_wdata[31:0]	00000000
s_axi_csr_wstrb[3:0]	0
s_axi_csr_wvalid	0
s_axi_csr_wready	0
s_axi_csr_bresp[1:0]	0
s_axi_csr_bvalid	0
s_axi_csr_bready	0
s_axi_csr_araddr[4:0]	00
s_axi_csr_arprot[2:0]	0
s_axi_csr_arvalid	0
s_axi_csr_arready	0
s_axi_csr_rdata[31:0]	00000000
s_axi_csr_rresp[1:0]	0
s_axi_csr_rvalid	0
s_axi_csr_rready	0
irq	0
s_axis_aclk	1
s_axis_aresetn	0
s_axis_ptctl_tdata[7:0]	00
s_axis_ptctl_tvalid	0
load_state	RESET_STATE
output_sel	0
t4_dds_load_offset[31:0]	00000000
t3_ramp_load_offset[15:0]	UUUU
m_axis_pd_testsig_tdata[15:0]	XXXX
m_axis_pd_testsig_tvalid	0
sample_period_ht	2500 ps
cntr_rst	0
arm_load	0
arm_clr	0
stay_armed	0
load_mode[2:0]	0
freq_val[31:0]	00000000
parameter_file_name[1:28]	"..\..\..\test_parameter
MODE_CNTL_REG[4:0]	00
FREQENCY_VAL_REG[4:0]	04
INTRPT_EN_REG[4:0]	14
INTRPT_STATUS_REG[4:0]	18
INTRPT_FLAG_REG[4:0]	1c

6.6 Synthesis and Implementation

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

This page is intentionally blank