IP CORE MANUAL

Sync Bus Interface for Higher Frequency Sample Clock IP

px_syncbus_intrfc1f

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

Manual Part Number: 807.48448 Rev: 1.0 – January 30, 2020

Manual Revision History

Date	Version	Comments
1/30/20	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™ Sync Bus Interface for Higher Frequency Sample Clock Core provides an interface to the front panel Sync Bus signals (gate, sync, and PPS signals) including timing signals from the user design. It also allows selection of the source of the output front panel Sync Bus signals, and the timing signals required to generate output timing event data.

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 Sync Bus Interface Core.

Features

- Generates timing event data streams and transfers them across the AXI4–Stream Interface
- Register access through AXI4–Lite Interface
- Accepts gate, sync, and PPS signals from the user design
- Differential termination can be applied to inputs based on the user design requirement
- Allows user to implement input buffers in low power or high performance mode
- Allows the selection of source of the output timing signals to the sync bus, and the timing signals which generate timing event data
- Generates interrupts for rising and falling edges of selected gate, sync, and PPS signals
- Includes an LED drive to indicate the status of the selected source
- Can be used with sample clock frequencies higher than 200 Mhz

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	N/A
Constraints File	Not Provided[b]
Simulation Model	N/A
Supported S/W Driver	HAL Software Support
Tested Design Flows	
Design Entry	Vivado® Design Suite 2019.2 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 Sync Bus Interface Core provides an interface to the front panel Sync Bus signals such as gate, sync and PPS signals and also supports generation of timing signals for the front panel Sync Bus and the user design. It has an AXI4–Lite Interface to access the Register Space within the core as shown in Figure 1–1. The Sync Bus Interface Core performs the following functions:

Gate:

- The Sync Bus Interface Core drives a differential LVDS gate output to the front panel Sync Bus based on the values assigned to Source Select Control Register bits. For more details about this register, refer to Section 4.4. This output is derived from one of the following sources:
  - Software generated gate signal (original and inverted signals)
  - Front panel LVTTL gate/trigger input (original and inverted signals)
  - LVTTL gate/trigger input from the front panel SSMC connector (original and inverted signals)
  - User gate signal input
- This core receives the differential LVDS gate and LVTTL gate/ trigger signals from the front panel Sync Bus.
- These incoming Gate signals are passed through an Input Buffer and Delay Module to add programmable tap delay to compensate for clock to gate skew, and allow for calibration of multi–board synchronization.
- This core also applies optional user–defined integer number of clock cycles of delay to the Sync Bus signal inputs to calibrate out long cable delays in multi–board synchronization.
- The Sync Bus Interface Core then allows the selection of the source of gate signal, which is used to generate timing event data for use within the user design, from the following sources:
  - Software generated gate
  - Front panel LVTTL gate/ trigger input (original and inverted signals)

- LVTTL gate/trigger input from the front panel SSMC connector (original and inverted signals)
- Front panel differential LVDS gate input (original and inverted signals)

1.1 Functional Description (continued)

Sync:

- The Sync Bus Interface Core drives a differential LVDS sync output to the front panel Sync Bus based on the values assigned to the Source Select Control Register bits. For more details about this register, refer to Section 4.4. This output is derived from one of the following sources:
  - Software generated sync signal (original and inverted signals)
  - Front panel LVTTL sync/PPS input (original and inverted signals)
  - Rising and falling edges of the selected gate signal which is used to generate the timing event data
  - User sync signal input
- The Sync Bus Interface Core receives the differential LVDS sync/PPS and LVTTL sync/PPS signals from the front panel Sync Bus.
- These incoming sync signals are passed through an Input Buffer and Delay Module to add programmable tap delay to compensate for clock to gate skew, and allow for calibration of multi–board synchronization.
- This core also applies optional user–defined integer number of clock cycles of delay to the Sync Bus signal inputs to calibrate out long cable delays in multi–board synchronization.
- The Sync Bus Interface Core then allows the selection of the source of sync signal, which is used to generate the timing event data for use within the user design, from the following sources:
  - Software generated sync signal
  - Front panel LVTTL sync/PPS input (original and inverted signals)
  - Rising and falling edges of the selected gate signal which is used to generate the timing event data
  - Front panel differential LVDS sync/ PPS input (original and inverted signals)

1.1 Functional Description (continued)

PPS:

- The Sync Bus Interface Core allows the selection of the source of PPS signal, which is used to generate the timing event data for timestamping purposes in the user design, from the following sources:
  - Software generated PPS signal
  - Front panel LVTTL sync/PPS input (original and inverted signals)
  - Rising and falling edges of the selected gate signal which is used to generate the timing event data
  - Front panel differential LVDS sync/PPS input (original and inverted signals)
  - Front Panel differential LVTTL gate/trigger input (original and inverted signals)
  - LVTTL gate/trigger input from the front panel SSMC connector (original and inverted signals)
  - User PPS signal input

The LVDS gate and sync outputs from the Sync Bus Interface Core are passed to the front panel Sync Bus when the Sync Bus Master Control output is enabled by setting the Source Select Control Register bit 0 to ‘1’. For more details about this register, refer to Section 4.4. This core generates interrupts for rising and falling edges of selected gate, sync and PPS signals. This core also generates an LED output to indicate the sta–tus of the source selected.

Figure 1–1 is a top–level block diagram of the Pentek Sync Bus Interface Core. The modules within the block diagram are explained in the later sections of this manual.

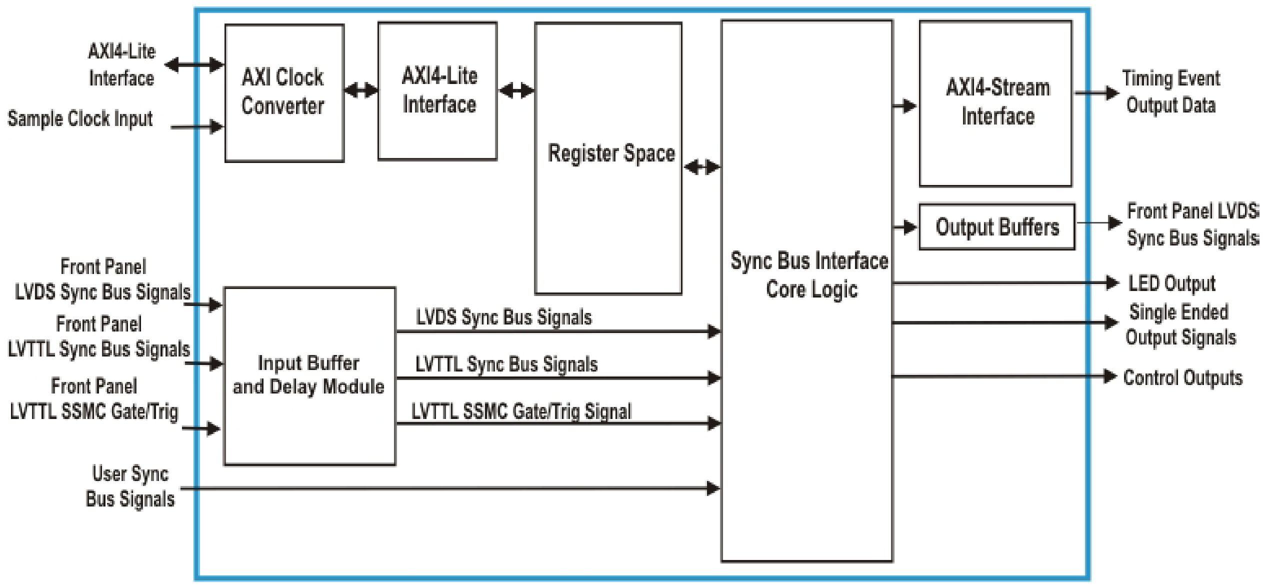
- **AXI Clock Converter Core:** The AXI Clock Converter Core is included in the Xilinx® AXI Interconnect Core and is used to connect one AXI memory–mapped slave to another AXI memory–mapped master which is operating in a different clock domain. In the Sync Bus Interface core, the AXI Clock Converter is used to operate the Register Space in the sample clock domain, which is the input clock to the core.

- **AXI4–Stream Interface:** The Sync Bus Interface Core has an AXI4–Stream Slave Interface to transfer timing event data streams through the output ports. For more details about the AXI4–Stream 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 of the core. For additional details about the AXI4–Lite Interface, refer to Section 3.1 AXI4–Lite Core Interfaces.

1.1 Functional Description (continued)

Figure 1–1: Sync Bus Interface Core Block Diagram



- **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.

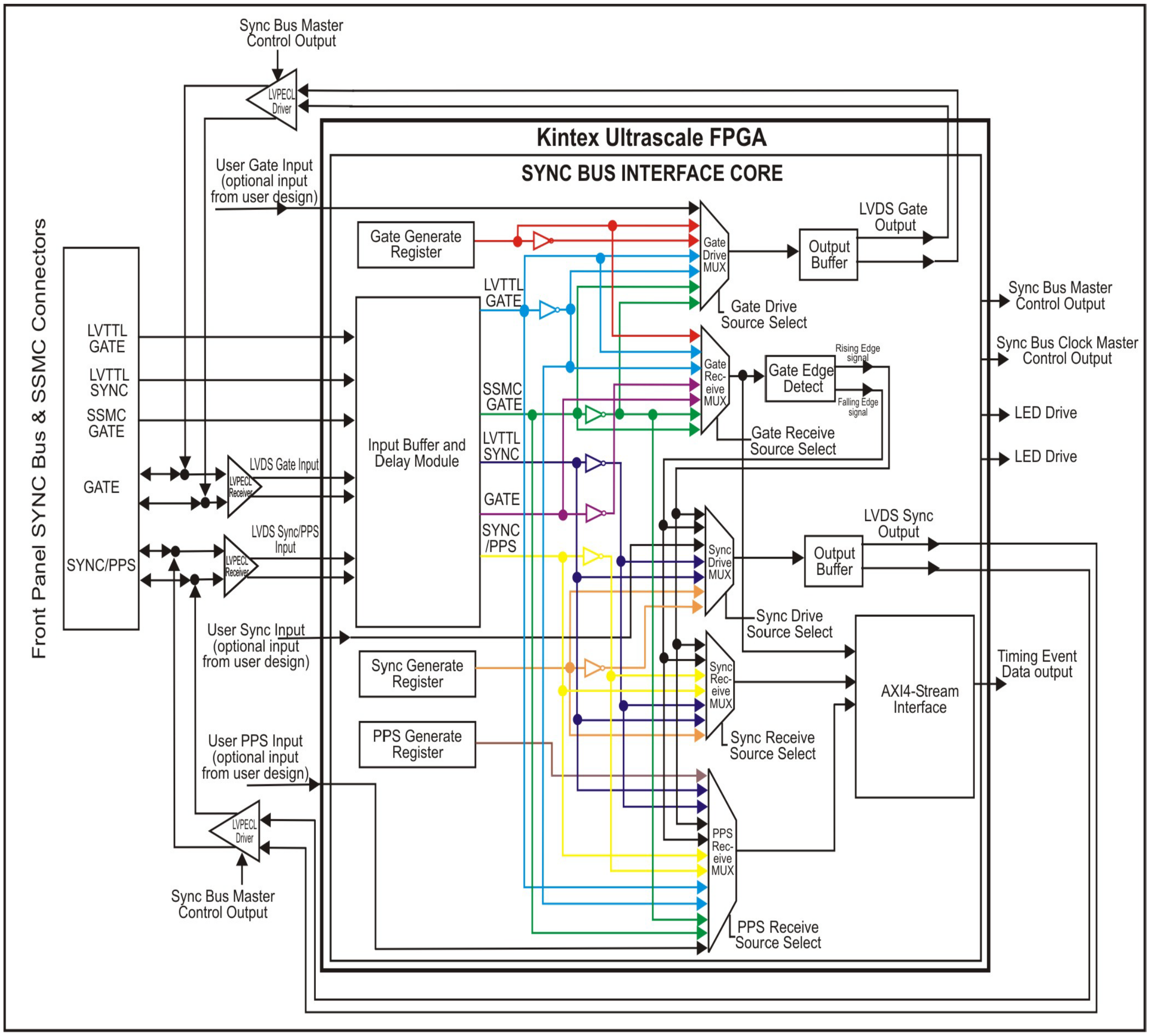
- **Sync Bus Interface Core Logic:** This module includes the required logic for selection of sources of the outgoing front panel Sync Bus signals, and timing signals to create the timing event data streams for user applications. The Figure 1–2 shows the Sync Bus Interface Core within the Kintex Ultrascale FPGA coupled with the external logic to illustrate signal flow from, and to, the front panel Sync Bus and the user design. It also shows the generation of LVDS gate and sync signal outputs and timing event data streams. The external logic includes LVPECL receivers to convert the incoming LVPECL Sync Bus signals into LVDS signals and LVPECL drivers to convert the output LVDS Sync Bus signals from the Sync Bus Interface Core into LVPECL signals. The LVPECL drivers are enabled through the Sync Bus master control output of this core.

- **Input Buffer and Delay Module:** This module includes input buffers for the incoming differential LVDS front panel Sync Bus signals. It also includes the Xilinx Input Delay Primitive to introduce programmable tap delay to the input signals. This core also introduces additional integer number of clock cycles of delay based on the values defined for the generic parameters as described in Section 2.5.

- **Output Buffers:** These are the output buffers for the Sync Bus signals to be transferred to the front panel Sync Bus.

1.1 Functional Description (continued)

Figure 1–2: Sync Bus Interface Core Logic Diagram

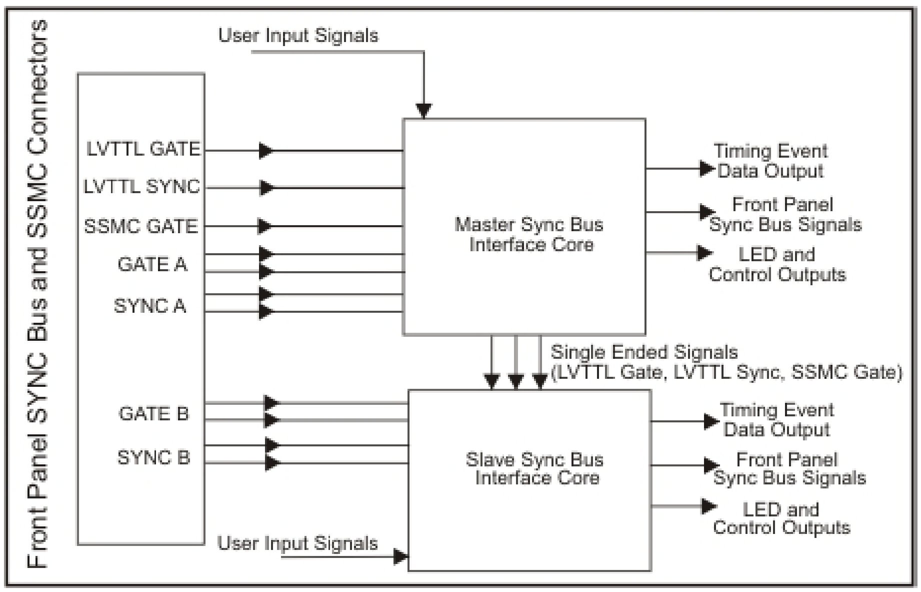


For user applications which include two modules that need access to the front panel Sync Bus LVTTL and SSMC signals, the user design can use two Sync Bus Interface Cores with one implemented as a Master and other as a Slave. In this case, the incoming LVTTL signals from the front panel are received by the Master and transmitted to the Slave as single ended output signals.

1.1 Functional Description (continued)

This application of the core enables access of the front panel Sync Bus signals to both modules within the user design. The Figure 1–3 shows two Sync Bus Interface Cores, one represented as a Master and other as a Slave for the front panel LVTTL and SSMC Sync Bus signals.

Figure 1–3: Sync Bus Interface Cores as Master and Slave



1.2 Applications

The Sync Bus Interface Core can be incorporated into any Kintex Ultrascale FPGA for use as an interface between the front panel Sync Bus and the rest of the user design. This core can be used for applications with sample clock frequencies higher than 200 Mhz.

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

Chapter 2: General Product Specifications

2.1 Standards

The Sync Bus Interface 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 Sync Bus Interface 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 Sync Bus Interface Core has two incoming clock signals. The input Sam–ple clock and the AXI4–Lite Interface clock both have 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 Sync Bus Interface 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	59
Flip–Flops	149

NOTE: Actual utilization may vary based on the user design in which the Sync Bus Interface 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 Sync Bus Interface 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
in_iodelay_grp	String	**IO Delay Group:** This is a string label that is used to group IODELAY tap delay components within the user design with the corresponding IDELAY control component.
is_single_ended_sig_ master	Boolean	**Is Single Ended Signal Master:** This parameter when set to True, indicates that the Sync Bus Interface Core is the Master and enables the generation of output single ended signals to another Slave Sync Bus Interface Core, from the input LVTTL signals to the core.
has_ssmc_gate_input		**Has SSMC Gate/Trigger Input:** This parameter indicates if the Sync Bus Interface Core has a gate/trigger input from the front panel SSMC connector.
has_user_pps_input		**Has User PPS Input:** This parameter is set to True when the Sync Bus Interface Core has a PPS signal input from the user design.
has_user_sync_drive_in		**Has User Sync Drive Input:** This parameter is set to True when the Sync Bus Interface Core has a user–defined source for the LVDS sync signal output to the front panel Sync Bus.
has_user_gate_drive_in		**Has User Gate Drive Input:** This parameter is set to True when the Sync Bus Interface Core has a user–defined source for the LVDS gate signal output to the front panel Sync Bus.
has_gate_trig_se_out		**Has Gate/Trigger Single Ended Output:** This parameter indicates if the Sync Bus Interface Core has a gate/ trigger single ended output. This parameter is valid only when the generic parameter **is_single_ended_sig_master** is set to True.
has_sync_pps_se_out		**Has Sync/PPS Single Ended Output:** This parameter indicates if the Sync Bus Interface Core has a sync/PPS single ended output. This parameter is valid only when the generic parameter **is_single_ended_sig_master** is set to True.
has_ssmc_gate_se_out		**Has SSMC gate Single Ended Output:** This parameter indicates if the Sync Bus Interface Core has a SSMC gate single ended output. This parameter is valid only when the generic parameter **is_single_ended_sig_master** is set to True.

Table 2-2: Generic Parameters (Continued)		
Port/Signal Name	**Type**	**Description**
initial_gate_tap_delay	Integer	**Initial Differential Gate Tap Delay at Reset:** This parameter defines the initial tap delay (in number of taps) to be introduced to the differential LVDS gate/trigger input from the front panel Sync Bus at reset. It can range from 0 to 511.
initial_sync_tap_delay		**Initial Differential Sync/PPS Tap Delay at Reset:** This parameter defines the initial tap delay (in number of taps) to be introduced to the differential LVDS sync/PPS input from the front panel Sync Bus at reset. It can range from 0 to 511.
initial_ttl_gate_tap_ delay	Integer	**Initial TTL Gate/Trigger Tap Delay at Reset:** This parameter defines the initial tap delay (in number of taps) to be introduced to the LVTTL gate/trigger input from the front panel Sync Bus at reset. It can range from 0 to 511.
initial_ttl_sync_tap_ delay		**Initial TTL Sync/PPS Tap Delay at Reset:** This parameter defines the initial tap delay (in number of taps) to be introduced to the LVTTL sync/PPS input from the front panel Sync Bus at reset. It can range from 0 to 511.
initial_ssmc_gate_tap _delay		**Initial SSMC Gate Tap Delay at Reset:** This parameter defines the initial tap delay (in number of taps) to be introduced to the gate/trigger input from the front panel SSMC connector at reset. It can range from 0 to 511.
initial_gate_int_delay		**Initial Differential Gate/Trigger Integer Delay at Reset:** This parameter defines the integer number of clock cycles of delay to be introduced to the differential LVDS gate/trigger input from the front panel Sync Bus at reset. It can range from 0 to 3.
initial_sync_int_delay		**Initial Differential Sync/PPS Integer Delay at Reset:** This parameter defines the integer number of clock cycles of delay to be introduced to the differential LVDS sync/PPS input from the front panel Sync Bus at reset. It can range from 0 to 3.
initial_ttl_gate_int_delay		**Initial LVTTL Gate/Trigger Integer Delay at Reset:** This parameter defines the integer number of clock cycles of delay to be introduced to the LVTTL gate/trigger input from the front panel Sync Bus at reset. It can range from 0 to 3.
initial_ttl_sync_int_ delay		**Initial LVTTL Sync/PPS Integer Delay at Reset:** This parameter defines the integer number of clock cycles of delay to be introduced to the LVTTL sync/PPS input from the front panel Sync Bus at reset. It can range from 0 to 3.
differential_term	Boolean	**Differential Termination:** This parameter is used to enable/ disable the internal 100Ω termination for the inputs. When set to True, differential termination is enabled.
ibuf_low_pwr		**Input Buffer Low Power:** Sets the input buffer performance. True = Input buffers implemented in low power mode; False = Input buffers implemented in high performance mode

Table 2-2: Generic Parameters (Continued)		
Port/Signal Name	**Type**	**Description**
idelaycntrl_refclk_freq	Integer	**IDelay Control Reference Clock:** It is the reference clock frequency in MHz for the input tap delay logic of this core. Typically this is 200 MHz.
initial_led_src		**Initial LED Source:** This parameter indicates the initial source of the LED at reset. 0 – Disabled 1 – Selected gate signal 2 – Selected sync signal 3 – Selected PPS Signal 4 – Output gate signal to front panel Sync Bus 5 – Output sync/PPS signal front panel Sync Bus 6 – User PPS logic signal
led_pulse_stretch		**LED Pulse Stretcher:** The LED pulse is stretched based on this value to make short overload events more visible on the LED. This value can range from 0 to 65535.

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
- I/O Signals

3.1 AXI4–Lite Core Interfaces

The Sync Bus Interface 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 Sync Bus Interface 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	6	**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 Sync Bus Interface Core. Note that the Register Space registers occupy an address range of [Base Address + (0x00 to 0x1C)].
s_axi_csr_awprot	Input	3	**Protection:** The Sync Bus Interface Core ignores these bits.

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

Port	Direction	Width	Description
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 Sync Bus Interface 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 Sync Bus Interface 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 Sync Bus Interface 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 Sync Bus Interface 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.

Table 3-1: Control/Status Register (CSR) Interface Port Descriptions (Continued)			
Port	**Direction**	**Width**	**Description**
s_axi_csr_bvalid	Output	1	**Write Response Valid:** This signal is asserted by the Sync Bus Interface 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	6	**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 Sync Bus Interface Core.
s_axi_csr_arprot	Input	3	**Protection:** These bits are ignored by the Sync Bus Interface 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 Sync Bus Interface 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 Sync Bus Interface 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 Sync Bus Interface 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 Sync Bus Interface 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 Sync Bus Interface Core has the following AXI4–Stream Interface to transfer data streams.

- Timing Events (PTCTL) Interface: The interface through which timing event 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 Master 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
m_axis_timecntl _tdata	Output	8	**Output Data:** This is timing event data and indicates the gate, sync, and PPS signal positions. tdata[0] – Gate positions tdata[1] – Sync positions tdata[2] – PPS positions
m_axis_timecntl _tvalid		1	**Output Data Valid:** Asserted when data is valid on **m_axis_ptctl_tdata**.

3.3 I/O Signals

The I/O port/signal descriptions of the top level module of the Sync Bus Interface Core are discussed in Table 3–3.

Table 3-3: I/O Signals

Port/Signal Name	Type	Direction	Description
gate_in_p	std_logic	Input	**Differential LVDS gate/trigger input from the front panel Sync Bus**
gate_in_n			
sync_pps_in_p			**Differential LVDS sync/PPS input from the front panel Sync Bus**
sync_pps_in_n			
gate_trig_ttl_in			**LVTTL gate/trigger input from the front panel Sync Bus**
sync_pps_ttl_in			**LVTTL sync/PPS input from the front panel Sync Bus**
ssmc_gate_ttl_in			**LVTTL gate input from the front panel SSMC connector:** This input signal is valid only when the generic parameter **has_ssmc_gate_input** is set to True.
user_pps_in			**User PPS Input:** This is an input PPS signal from the user design and is valid only when the generic parameter **has_user_pps_input** is set to True.
user_sync_drv_in			**User Sync Drive Input:** This is the user–defined sync input which acts as a source in the generation of the sync output signal to the front panel Sync Bus. This signal is valid only when the generic parameter **user_sync_ drive_in** is set to True.
user_gate_drv_in			**User Gate Drive Input:** This the user defined gate input which acts as a source in the generation of the Gate output signal to the Front panel Sync Bus.This signal is valid only when the generic parameter **user_gate_ drive_in** is set to True.
sample_clk			**Sample Clock Input:** This is the sample clock input of the core.
gate_out_p		Output	**Differential LVDS gate/trigger output to the front panel Sync Bus**
gate_out_n			
sync_pps_out_p			**Differential LVDS sync/PPS output to the front panel Sync Bus**
sync_pps_out_n			

Table 3-3: I/O Signals (Continued)			
Port/Signal Name	**Type**	**Direction**	**Description**
gate_trig_se_out	std_logic	Output	**Single Ended Gate/Trigger output signal to the Slave:** This output signal is generated when both the generic parameters **is_single_ended_sig_master** and **has_gate_trig_se_out** are set to True.
sync_pps_se_out			**Single Ended Sync/PPS output signal to the Slave:** This output signal is generated when both the generic parameters **is_single_ ended_sig_master** and **has_sync_pps_se_out** are set to True.
ssmc_gate_se_out			**Single Ended SSMC Gate/Trigger output signal to the Slave:** This output signal is generated when both the generic parameters **is_single_ended_sig_master** and **has_ssmc_gate_se_out** are set to True.
led_n			**LED Drive:** This parameter indicates the status of the source of the LED. The source of LED at reset is defined using the generic parameter **initial_led_src**.
sbus_master_n			**Sync Bus Master:** Active Low. When Low, this control output enables the external LVPECL drivers to drive the LVDS Sync Bus Output signals from the Sync Bus Interface Core to the front panel Sync Bus.
clk_master_n			**Sync Bus Clock Master:** Active Low. When Low, this control output enables the external clock buffer to output a clock signal to the front panel Sync Bus.

Chapter 4: Register Space

This chapter provides the memory map and register descriptions for the register space of the Sync Bus Interface Core. The memory map is provided in Table 4–1.

Table 4–1: Register Space Memory Map

Register Name	Address (Base Address +)	Access	Description
Gate Receive Buffer Control	0x00	R/W	Controls the tap delay and integer delay of the LVDS and LVTTL gate input signals.
Sync Receive Buffer Control	0x04		Controls the tap delay and integer delay of the LVDS and LVTTL sync/pps input signals.
Auxiliary Receive Buffer Control	0x08		Controls the tap delay and integer delay of the SSMC gate/trigger input signal.
Source Select Control	0x0C		Controls the driver and receiver sources of the Sync Bus signals, and the source of the LED.
Gate Generate	0x10		Controls the software generated gate signal.
Sync Generate	0x14		Controls the software generated sync signal.
PPS Generate	0x18		Controls the software generated PPS signal.
Status Register	0x1C	R	Indicates the status of the selected sources of the front panel Sync Bus output signals and the input differential LVDS and LVTTL signals
Interrupt Enable Register	0x20	R/W	Interrupt enable bits
Interrupt Status Register	0x24	R	Interrupt source status bits
Interrupt Flag Register	0x28	R/W	Interrupt flag bits

4.1 Gate Receive Buffer Control Register

This register controls the tap delay and integer delay of the LVDS and LVTTL gate/trigger input signals, from the front panel Sync Bus, when they are passed through the Input Buffer and Delay module of the Sync Bus Interface Core. The Gate Receive Buffer Control Register is illustrated in Figure 4–1 and described in Table 4–2.

NOTE: The generic parameters defined by the user are taken as default values for the regiater bits.

Figure 4–1: Gate Receive Buffer Control Register

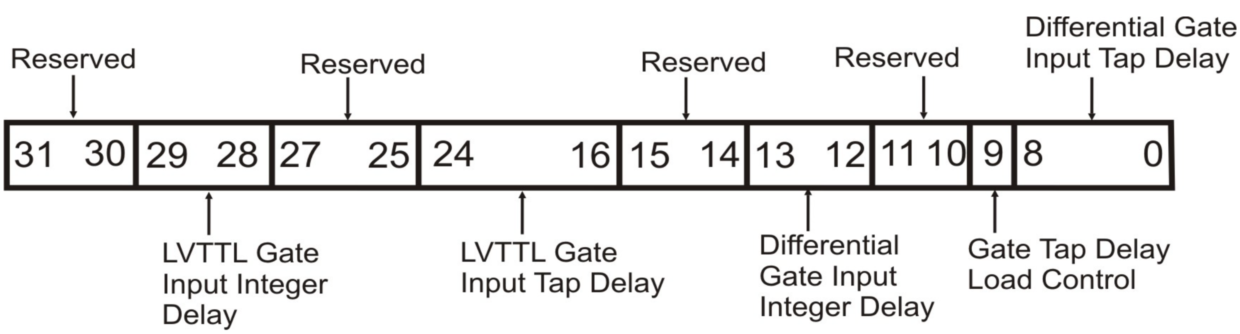


Table 4–2: Gate Receive Buffer Control Register (Base Address + 0x00)

Bits	Field Name	Default Value	Access Type	Description
31:30	Reserved	N/A	N/A	**Reserved**
29:28	gate_ttl_int_delay	–	R/W	**LVTTL Gate/Trigger Input Integer Delay:** This is the integer number of clock cycles of delay to be introduced into the LVTTL gate/ trigger input from the front panel Sync Bus after passing through the tap delay logic.
27:25	Reserved	N/A	N/A	**Reserved**
24:16	ttlgate_tap_delay	–	R/W	**LVTTL Gate/Trigger Input Tap Delay:** These bits are used to control the number of taps required to introduce the desired delay to the LVTTL gate/trigger input from the front panel Sync Bus. The Kintex Ultrascale FPGA has a 512 tap delay line with a maximum delay value of 7600ps and minimum delay value of 1250ps. This value can range from 0 to 511.
15:14	Reserved	N/A	N/A	**Reserved**

Table 4–2: Gate Receive Buffer Control Register (Base Address + 0x00) (Continued)				
Bits	**Field Name**	**Default Value**	**Access Type**	**Description**
13:12	gate_int_delay	–	R/W	**Differential LVDS Gate/Trigger Input Integer Delay:** This is the integer number of clock cycles of delay introduced into the LVDS gate/trigger input from the front panel Sync Bus after passing through the tap delay logic.
11:10	Reserved	N/A	N/A	**Reserved**
9	gate_dly_ld_ctl	0	R/W	**Gate Tap Delay Load Control:** This bit controls loading of the LVDS gate/trigger input tap delay value (bits [8:0]) and LVTTL gate/trigger input tap delay value (bits [24:16]) into their corresponding Xilinx IDELAY components of the core. The IDELAY components are used to introduce tap delay to the LVDS gate/trigger input and LVTTL gate/trigger input from the front panel Sync Bus. Toggling this bit ‘1’ then ‘0’ will enable the tap delay value to be loaded into the IDELAY components.
8:0	gate_tap_delay	–	R/W	**Differential LVDS Gate/Trigger Input Tap Delay:** These bits are used to control the number of taps required to introduce the desired delay to the LVDS gate/trigger input from the front panel Sync Bus. The Kintex Ultrascale FPGA has a 512 tap delay line with a maximum delay value of 7600ps and minimum delay value of 1250ps. This value can range from 0 to 511.

4.2 Sync Receive Buffer Control Register

This register controls the tap delay and integer delay of the LVDS and LVTTL sync/PPS input signals, from the front panel Sync Bus, when they are passed through the Input Buffer and Delay module of the Sync Bus Interface Core. The Sync Receive Buffer Control Register is illustrated in Figure 4–2 and described in Table 4–3.

NOTE: The generic parameters defined by the user are taken as default values for the regiater bits.

Figure 4–2: Sync Receive Buffer Control Register

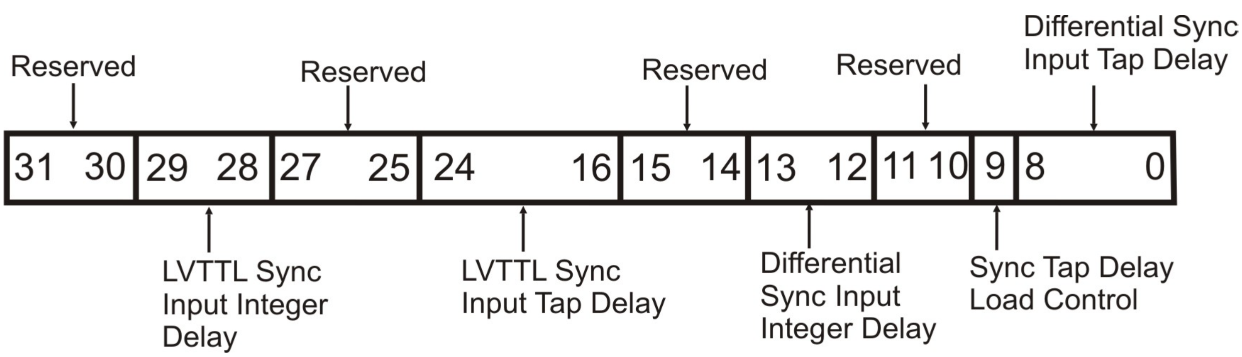


Table 4–3: Sync Receive Buffer Control Register (Base Address + 0x04)

Bits	Field Name	Default Value	Access Type	Description
31:30	Reserved	N/A	N/A	**Reserved**
29:28	sync_ttl_int_delay	–	R/W	**LVTTL Sync/PPS Input Integer Delay:** This is the integer number of clock cycles of delay to be introduced into the LVTTL sync/PPS input from the front panel Sync Bus after passing through the tap delay logic.
27:25	Reserved	N/A	N/A	**Reserved**
24:16	ttlsync_tap_delay	–	R/W	**LVTTL Sync/PPS Input Tap Delay:** These bits are used to control the number of taps required to introduce the desired delay to the LVTTL sync/PPS input from the front panel Sync Bus. The Kintex Ultrascale FPGA has a 512 tap delay line with a maximum delay value of 7600ps and minimum delay value of 1250ps. This value can range from 0 to 511.
15:14	Reserved	N/A	N/A	**Reserved**

Table 4–3: Sync Receive Buffer Control Register (Base Address + 0x04) (Continued)				
Bits	**Field Name**	**Default Value**	**Access Type**	**Description**
13:12	sync_int_delay	–	R/W	**Differential LVDS Sync/PPS Input Integer Delay:** This is the integer number of clock cycles of delay introduced into the LVDS sync/PPS input from the front panel Sync Bus, after passing through the tap delay logic.
11:10	Reserved	N/A	N/A	**Reserved**
9	sync_dly_ld_ctl	0	R/W	**Sync Tap Delay Load Control:** This bit controls loading of the LVDS sync/PPS input tap delay value (bits [8:0]) and LVTTL sync/PPS input tap delay value (bits [24:16]) into their corresponding Xilinx IDELAY components of the core. The IDELAY components are used to introduce tap delay to the LVDS sync/ PPS input and LVTTL sync/PPS input from the front panel Sync Bus. Toggling this bit '1' then '0' will enable the tap delay value to be loaded into the IDELAY components.
8:0	sync_tap_delay	–		**Differential LVDSSync/PPS Input Tap Delay:** These bits are used to control the number of taps required to introduce the desired delay to the LVDS sync/PPS input from the front panel Sync Bus. The Kintex Ultrascale FPGA has a 512 tap delay line with a maximum delay value of 7600ps and minimum delay value of 1250ps. This value can range from 0 to 511.

4.3 Auxiliary Receive Buffer Control Register

This register controls the tap delay and integer delay of the LVTTL gate/trigger input signal, from the front panel SSMC connector, when they are passed through the Input Buffer and Delay module of the Sync Bus Interface Core. The Auxiliary Receive Buffer Control Register is illustrated in Figure 4–3 and described in Table 4–4.

NOTE: The generic parameters defined by the user are taken as default values for the regiater bits.

Figure 4–3: Auxiliary Receive Buffer Control Register

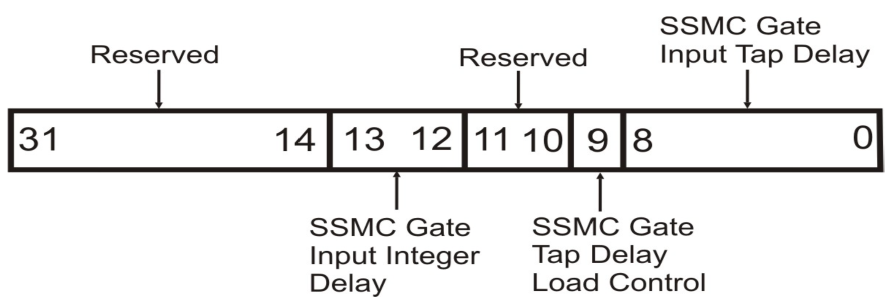


Table 4–4: Auxiliary Receive Buffer Control Register (Base Address + 0x08)

Bits	Field Name	Default Value	Access Type	Description
31:14	Reserved	N/A	N/A	**Reserved**
13:12	ssmc_ttl_int_delay	–	R/W	**SSMC LVTTL Gate/Trigger Input Integer Delay:** This is the integer number of clock cycles of delay introduced into the LVTTL gate/trigger input from the front panel SSMC connector after passing through the tap delay logic.
11:10	Reserved	N/A	N/A	**Reserved**

Table 4–4: Auxiliary Receive Buffer Control Register (Base Address + 0x08) (Continued)

Bits	Field Name	Default Value	Access Type	Description
9	ssmc_dly_ld_ctl	0	R/W	**SSMC Gate Tap Delay Load Control:** This bit controls loading of the SSMC gate/trigger input tap delay value (bits [8:0]) into the Xilinx IDELAY component of the core. The IDELAY component is used to introduce tap delay to the LVTTL gate/trigger input from the front panel SSMC connector. Toggling this bit '1' then '0' will enable the tap delay value to be loaded into the IDELAY component.
8:0	ttlssmc_tap_delay	–		**SSMC LVTTL Gate/Trigger Input Tap Delay:** These bits are used to control the number of taps required to introduce the desired delay to the LVTTL gate/trigger input from the front panel SSMC connector. The Kintex Ultrascale FPGA has a 512 tap delay line with a maximum delay value of 7600ps and minimum delay value of 1250ps. It can range from 0 to 511.

4.4 Source Select Control Register

This register controls the selection of sources of the input and output Sync Bus signals, and the LED. It also controls the Sync Bus master and clock master control outputs of the Sync Bus Interface Core. The Source Select Control Register is illustrated in Figure 4–4 and described in Table 4–5.

NOTE: The generic parameters defined by the user are taken as default values for the regiater bits.

Figure 4–4: Source Select Control Register

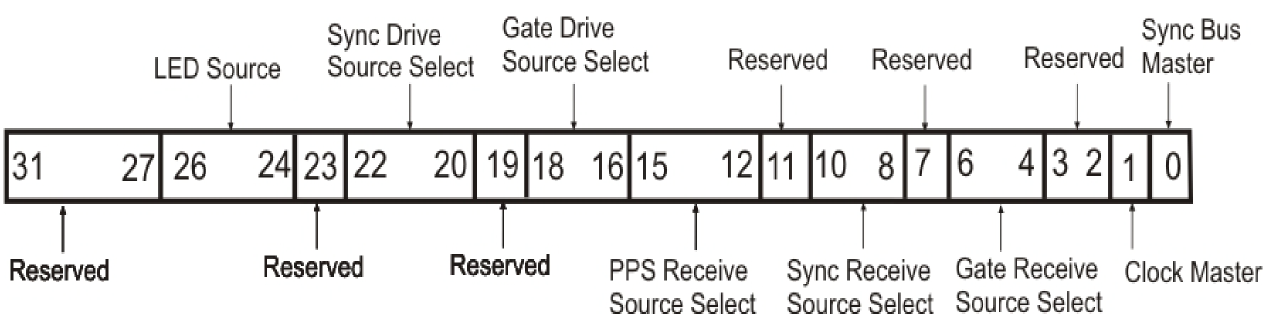


Table 4–5: Source Select Control Register (Base Address + 0x0C)

Bits	Field Name	Default Value	Access Type	Description
31:27	Reserved	N/A	N/A	**Reserved**
26:24	led_src	–	R/W	**LED Source:** These bits control the source of the LED. 000 – Disabled 001 – Selected gate input 010 – Selected sync input 011 – Selected PPS input 100 – Output gate to the front panel Sync Bus 101 – Output sync/PPS to the front panel Sync Bus 110 & 111 – User PPS input
23	Reserved	N/A	N/A	**Reserved**

Table 4–5: Source Select Control Register (Base Address + 0x0C) (Continued)				
Bits	**Field Name**	**Default Value**	**Access Type**	**Description**
22:20	sync_drv_src	000	R/W	**Sync Drive Source Select:** These bits select the source of the sync signal output of the Sync Bus Interface Core to the front panel Sync Bus. 000 – Sync signal from Sync Generate Register (Active High) 001 – Inverted sync signal from Sync Generate Register (Active Low) 010 – Front panel LVTTL sync/PPS input (As received) 011 – Front panel LVTTL sync/PPS input (Inverted) 100 – Rising Edge of selected gate signal output 101 – Falling Edge of selected gate signal output 110 & 111 – User sync input
19	Reserved	N/A	N/A	**Reserved**
18:16	gate_drv_src	000	R/W	**Gate Drive Source Select:** These bits select the source of the gate signal output of the Sync Bus Interface Core to the Front Panel Sync Bus. 000 – Gate signal from Gate Generate Register (Active High) 001 – Inverted gate signal from Gate Generate Register (Active Low) 010 – Front panel LVTTL gate/trigger input (As received) 011 – Front panel LVTTL gate/trigger input (Inverted) 100 – Front panel LVTTL gate/trigger SSMC input (As received) 101 – Front panel LVTTL gate/trigger SSMC input (Inverted) 110 & 111 – User gate input

Table 4–5: Source Select Control Register (Base Address + 0x0C) (Continued)				
Bits	**Field Name**	**Default Value**	**Access Type**	**Description**
15:12	pps_rcv_src	0000	R/W	**PPS Receive Source Select:** These bits select the source of the PPS signal which generates the timing event data that is used within the user design. When there are no incoming signals from the SSMC connector or user inputs the corresponding sources are set to ‘0’. 0000 – None (Off) 0001 – PPS signal from the PPS Generate Register 0010 – Front panel LVTTL sync/PPS input (As received) 0011 – Front panel LVTTL sync/PPS input (Inverted) 0100 – Rising edge of gate signal from the selected gate receive source 0101 – Falling edge of gate signal from the selected gate receive source 0110 – Front panel differential sync/PPS signal input (As received) 0111 – Front panel differential sync/PPS signal input (Inverted) 1000 – Front panel LVTTL gate/trigger input (As received) 1001 – Front panel LVTTL gate/trigger input (Inverted) 1010 – Front panel LVTTL gate/trigger SSMC input (As received) 1011 – Front panel LVTTL Gate/Trigger SSMC Input (Inverted) 1100 – User PPS input others – ‘0’
11	Reserved	N/A	N/A	**Reserved**
10:8	sync_rcv_src	000	R/W	**Sync Receive Source Select:** These bits select the source of the sync signal which generates the timing event data, which is used within the user design. 000 – None (Off) 001 – Sync signal from Sync Generate Register 010 – Front panel LVTTL sync/PPS signal input (As received) 011 – Front panel LVTTL sync/PPS signal input (Inverted) 100 – Rising edge of gate signal from the selected gate receive source 101 – Falling edge of gate signal from the selected gate receive source 110 – Front panel differential sync/PPS input (As received) 111 – Front panel differential sync/PPS input (Inverted)
7	Reserved	N/A	N/A	**Reserved**

Table 4–5: Source Select Control Register (Base Address + 0x0C) (Continued)				
Bits	**Field Name**	**Default Value**	**Access Type**	**Description**
6:4	gate_rcv_src	000	R/W	**Gate Receive Source Select:** These bits select the source of the Gate Signal which generates the Timing event data, which is used within the user design. When there is no incoming signal from the SSMC connector the corresponding source is set to ‘0’. 000 – None (Off) 001 – Gate signal from Gate Generate Register 010 – Front panel LVTTL gate/trigger signal input (As received) 011 – Front panel LVTTL gate/trigger signal input (Inverted) 100 – Front panel LVTTL gate/trigger SSMC input (As received) 101 – Front panel LVTTL gate/trigger SSMC input (Inverted) 110 – Front panel differential gate/trigger input (As received) 111 – Front panel differential gate/trigger input (Inverted)
3:2	Reserved	N/A	N/A	**Reserved**
1	clk_master	0	R/W	**Sync Bus Clock Master:** This bit enables/ disables the Sync Bus Interface Core to operate as the clock master on the front panel Sync Bus. When this bit is set to ‘1’, this core drives a clock signal to the front panel Sync Bus. 0 = Slave 1 = Master
0	sbus_master			**Sync Bus Master:** This bit enables/ disables the Sync Bus Interface Core to operate as the bus master on the front panel Sync Bus. When this bit is set to ‘1’, this core drives gate and sync signals to the front panel Sync Bus. 0 = Slave 1 = Master

4.5 Gate Generate Register

This register controls the generation of the gate signal in the Sync Bus interface Core. When the gate source select bits are set to Gate Generate Register, the gate bit in this register creates a gate signal for the user design, and the front panel Sync Bus. The Gate Generate Register is illustrated in Figure 4–5 and described in Table 4–6.

NOTE: The generic parameters defined by the user are taken as default values for the regiater bits.

Figure 4–5: Gate Generate Register

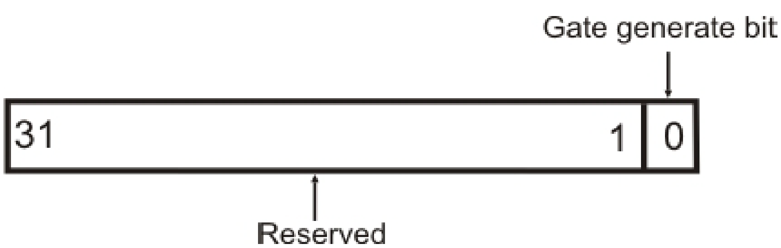


Table 4–6: Gate Generate Register (Base Address + 0x10)				
Bits	**Field Name**	**Default Value**	**Access Type**	**Description**
31:1	Reserved	N/A	N/A	**Reserved**
0	gate_reg_gen	0	R/W	**Gate**: 0 = Inactive 1 = Active

4.6 Sync Generate Register

This register controls the generation of the sync signal in the Sync Bus interface Core. When the sync source select bits are set to Sync Generate Register, the sync bit in this register creates a sync signal for the user design, and the front panel Sync Bus. The Sync Generate Register is illustrated in Figure 4–6 and described in Table 4–7.

NOTE: The generic parameters defined by the user are taken as default values for the regiater bits.

Figure 4–6: Sync Generate Register

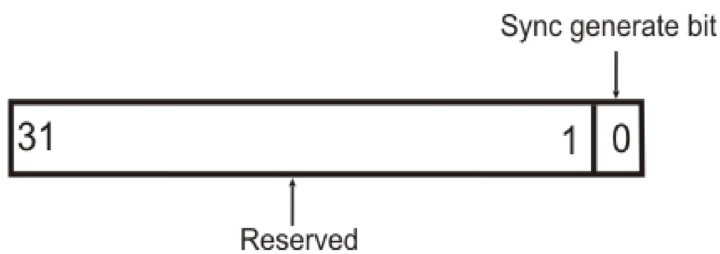


Table 4–7: Sync Generate Register (Base Address + 0x14)

Bits	Field Name	Default Value	Access Type	Description
31:1	Reserved	N/A	N/A	**Reserved**
0	sync_reg_gen	0	R/W	**Sync**: 0 = Inactive 1 = Active

4.7 PPS Generate Register

This register controls the generation of the PPS signal in the Sync Bus interface Core. When the PPS source select bits are set to PPS Generate Register, the PPS bit in this register creates a PPS signal for the user design, and the front panel Sync Bus. The PPS Generate Register is illustrated in Figure 4–7 and described in Table 4–8.

NOTE: The generic parameters defined by the user are taken as default values for the regiater bits.

Figure 4–7: PPS Generate Register

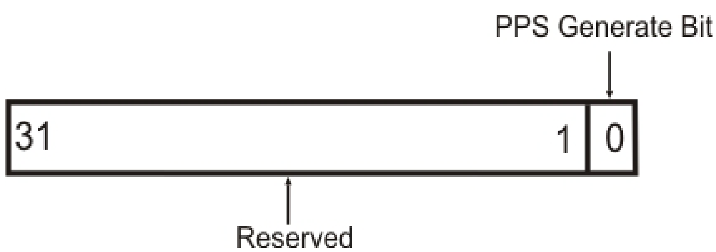


Table 4–8: PPS Generate Register (Base Address + 0x18)				
Bits	**Field Name**	**Default Value**	**Access Type**	**Description**
31:1	Reserved	N/A	N/A	**Reserved**
0	pps_reg_gen	0	R/W	**PPS**: 0 = Inactive 1 = Active

4.8 Status Register

This register indicates the status of the selected gate, sync, and PPS signals used for generation of timing events. It also indicates the SSMC gate input, LVTTL gate/trigger input, LVTTL sync/PPS input, LVDS gate/ trigger input, and LVDS sync/PPS input. The Status Register is illustrated in Figure 4–8 and described in Table 4–9.

Figure 4–8: Status Register

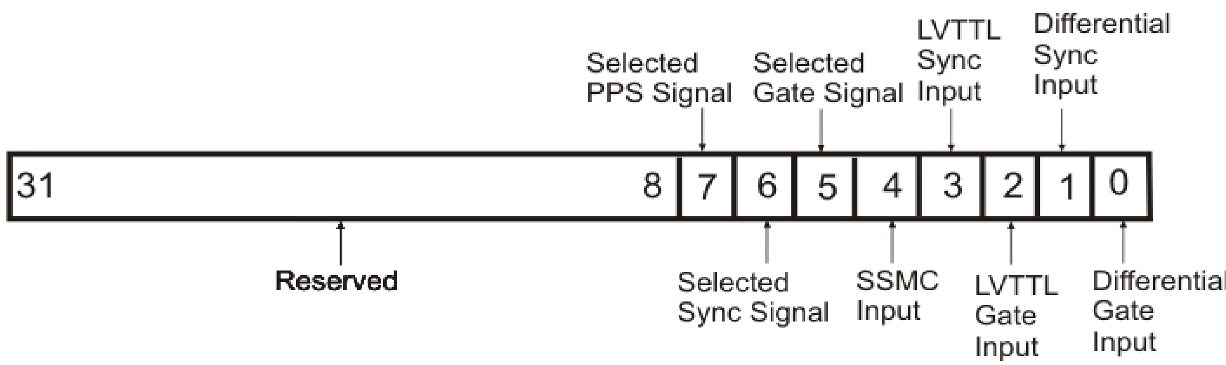


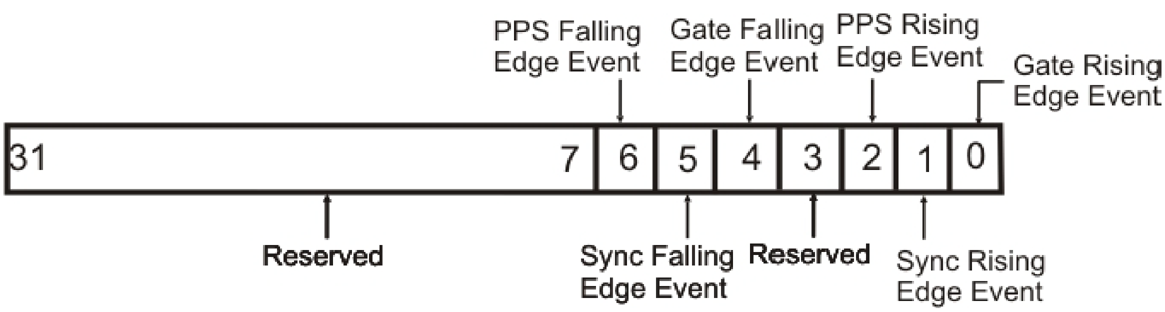
Table 4–9: Status Register (Base Address + 0x1C)				
Bits	**Field Name**	**Default Value**	**Access Type**	**Description**
31:8	Reserved	N/A	N/A	**Reserved**
7	selected_pps	0	R	**Selected PPS Signal:** This bit indicates the status of the selected PPS signal, which is used to generate the timing event data output of the Sync Bus Interface Core. 0 = Inactive 1 = Active
6	selected_sync			**Selected Sync Signal:** This bit indicates the status of the selected sync signal, which is used to generate the timing event data output of the Sync Bus Interface Core. 0 = Inactive 1 = Active
5	selected_gate			**Selected Gate Signal:** This bit indicates the status of the selected gate signal, which is used to generate the timing event data output of the Sync Bus Interface Core. 0 = Inactive 1 = Active

Table 4–9: Status Register (Base Address + 0x1C) (Continued)				
Bits	**Field Name**	**Default Value**	**Access Type**	**Description**
4	ttl_ssmc_pin	0	R	**LVTTL Gate/ Trigger SSMC Input:** This bit indicates the status of the gate/trigger input from the front panel SSMC connector. 0 = Inactive 1 = Active
3	tt_sync_pin			**LVTTL Sync/PPS Input:** This bit indicates the status of the LVTTL sync/PPS input from the front panel Sync Bus. 0 = Inactive 1 = Active
2	tt_gate_pin			**LVTTL Gate/Trigger Input:** This bit indicates the status of the LVTTL gate/trigger input from the front panel Sync Bus. 0 = Inactive 1 = Active
1	diff_sync_pin			**Differential LVDS Sync/PPS Input:** This bit indicates the status of the LVDS sync/PPS input from the front panel Sync Bus. 0 = Inactive 1 = Active
0	diff_gate_pin			**Differential LVDS Gate/Trigger Input:** This bit indicates the status of the LVDS gate/trigger input from the front panel Sync Bus. 0 = Inactive 1 = Active

4.9 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 4.10). This register is illustrated in Figure 4–9 and described in Table 4–10.

Figure 4–9: Interrupt Enable Register



Bits	Field Name	Default Value	Access Type	Description
31:7	Reserved	N/A	N/A	**Reserved**
6	pps_fe_event	0	R/W	**PPS Signal Falling Edge Event:** This bit enables/ disables the PPS signal falling edge interrupt source of the selected PPS signal which is used to generate timing event data output. 0 = Disable interrupt 1 = Enable interrupt
5	sync_fr_event			**Sync Signal Falling Edge Event:** This bit enables/ disables the sync signal falling edge interrupt source of the selected sync signal which is used to generate timing event data output. 0 = Disable interrupt 1 = Enable interrupt
4	gate_fr_event			**Gate Signal Falling Edge Event:** This bit enables/ disables the gate signal falling edge interrupt source of the selected gate signal which is used to generate timing event data output. 0 = Disable interrupt 1 = Enable interrupt

Table 4–10: Interrupt Enable Register (Base Address + 0x20)

Table 4–10: Interrupt Enable Register (Base Address + 0x20) (Continued)				
Bits	**Field Name**	**Default Value**	**Access Type**	**Description**
3	Reserved	N/A	N/A	**Reserved**
2	pps_re_event	0	R/W	**PPS Signal Rising Edge Event:** This bit enables/disables the PPS signal rising edge interrupt source of the selected PPS signal which is used to generate timing event data output. 0 = Disable interrupt 1 = Enable interrupt
1	sync_re_event			**Sync Signal Rising Edge Event:** This bit enables/disables the sync signal rising edge interrupt source of the selected sync signal which is used to generate timing event data output. 0 = Disable interrupt 1 = Enable interrupt
0	gate_re_event			**Gate Signal Rising Edge Event:** This bit enables/disables the gate signal rising edge interrupt source of the selected gate signal which is used to generate timing event data output. 0 = Disable interrupt 1 = Enable interrupt

4.10 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 4–10 and described in Table 4–11.

Figure 4–10: Interrupt Status Register

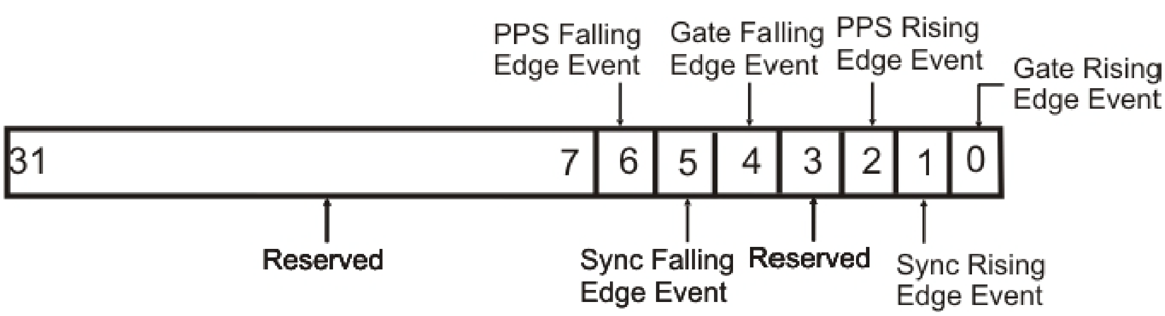


Table 4–11: Interrupt Status Register (Base Address + 0x24)				
Bits	**Field Name**	**Default Value**	**Access Type**	**Description**
31:7	Reserved	N/A	N/A	**Reserved**
6	pps_fe_event	0	R	**PPS Signal Falling Edge Event:** This bit indicates the status of the PPS signal falling edge interrupt source of the selected PPS signal which is used to generate timing event data output. 0 = Disable interrupt 1 = Enable interrupt
5	sync_fr_event			**Sync Signal Falling Edge Event:** This bit indicates the status of the sync signal falling edge interrupt source of the selected sync signal which is used to generate timing event data output. 0 = No interrupt 1 = Interrupt condition asserted

Table 4–11: Interrupt Status Register (Base Address + 0x24) (Continued)				
Bits	**Field Name**	**Default Value**	**Access Type**	**Description**
4	gate_fr_event	0	R	**Gate Signal Falling Edge Event:** This bit indicates the status of the gate signal falling edge interrupt source of the selected gate signal which is used to generate timing event data output. 0 = No interrupt 1 = Interrupt condition asserted
3	Reserved	N/A	N/A	**Reserved**
2	pps_re_event	0	R	**PPS Signal Rising Edge Event:** This bit indicates the status of the PPS signal rising edge interrupt source of the selected PPS signal which is used to generate timing event data output. 0 = No interrupt 1 = Interrupt condition asserted
1	sync_re_event			**Sync Signal Rising Edge Event:** This bit indicates the status of the sync signal rising edge interrupt source of the selected sync signal which is used to generate timing event data output. 0 = No interrupt 1 = Interrupt condition asserted
0	gate_re_event			**Gate Signal Rising Edge Event:** This bit indicates the status of the gate signal rising edge interrupt source of the selected gate signal which is used to generate timing event data output. 0 = No interrupt 1 = Interrupt condition asserted

4.11 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 4–11 and described in Table 4–12.

Figure 4–11: Interrupt Flag Register

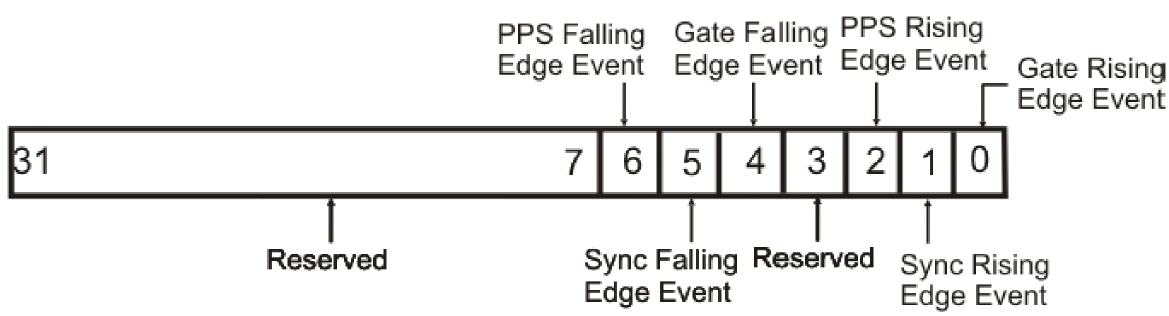


Table 4–12: Interrupt Flag Register (Base Address + 0x28)				
Bits	**Field Name**	**Default Value**	**Access Type**	**Description**
31:7	Reserved	N/A	N/A	**Reserved**

Table 4–12: Interrupt Flag Register (Base Address + 0x28) (Continued)				
Bits	**Field Name**	**Default Value**	**Access Type**	**Description**
6	pps_fe_event	0	R/Clr	**PPS Signal Falling Edge Event:** This bit indicates the PPS signal falling edge interrupt flag of the selected PPS signal which is used to generate timing event data output. **Read:** 0 = No interrupt 1 = Interrupt latched **Clear:** 1 = Clear latch
5	sync_fr_event			**Sync Signal Falling Edge Event:** This bit indicates the sync signal falling edge interrupt flag of the selected sync signal which is used to generate timing event data output. **Read:** 0 = No interrupt 1 = Interrupt latched **Clear:** 1 = Clear latch
4	gate_fr_event	0	R/Clr	**Gate Signal Falling Edge Event:** This bit indicates the gate signal falling edge interrupt flag of the selected gate signal which is used to generate timing event data output. **Read:** 0 = No interrupt 1 = Interrupt latched **Clear:** 1 = Clear latch
3	Reserved	N/A	N/A	**Reserved**
2	pps_re_event	0	R/Clr	**PPS Signal Rising Edge Event:** This bit indicates the PPS signal rising edge interrupt flag of the selected PPS signal which is used to generate timing event data output. **Read:** 0 = No interrupt 1 = Interrupt latched **Clear:** 1 = Clear latch
1	sync_re_event			**Sync Signal Rising Edge Event:** This bit indicates the sync signal rising edge interrupt flag of the selected sync signal which is used to generate timing event data output. **Read:** 0 = No interrupt 1 = Interrupt latched **Clear:** 1 = Clear latch
0	gate_re_event			**Gate Signal Rising Edge Event:** This bit indicates the gate signal rising edge interrupt flag of the selected gate signal which is used to generate timing event data output. **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 Sync Bus Interface Core.

5.1 General Design Guidelines

The Sync Bus Interface Core provides the required logic to generate Timing Event streams for the user design and Sync Bus output signals to the front panel Sync Bus. This IP core supports AXI4–Lite and AXI4–Stream user interfaces. The user can customize the core by setting the generic parameters as described in Section 2.5, and the control registers as described in Chapter 4.

NOTE: This core (px_syncbus_intrfc1f) is identical to px_syncbus_intrfc except that this core can be used for applications with sample clock frequencies higher than 200 Mhz.

5.2 Clocking

Sample Clock: **`sample_clk`**

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 Sync Bus Interface Core in the Sample Clock domain.

5.3 Resets

Main reset: **`s_axi_csr_aresetn`**

This is an active low reset synchronous with **`s_axi_csr_clk`**. When asserted, all FIFOs are flushed and all the control registers are cleared back to their initial default states.

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 Reg–ister 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/ status 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 data is transferred through the output ports to the user design. It is an AXI4–Stream Master Interface. For more details about this interface, refer to Section 3.2.1.

5.6 Programming Sequence

This section briefly describes the programming sequence of registers in the Sync Bus Interface Core.

1) Ensure that the Interrupt Flag Register is cleared.
2) Enable the interrupt enable bits based on the user design requirement.
3) Set the control registers with the required values.
4) Observe the outputs across the outputs ports.
5) When done check the Interrupt Flag Register and clear the interrupts.

5.7 Timing Diagrams

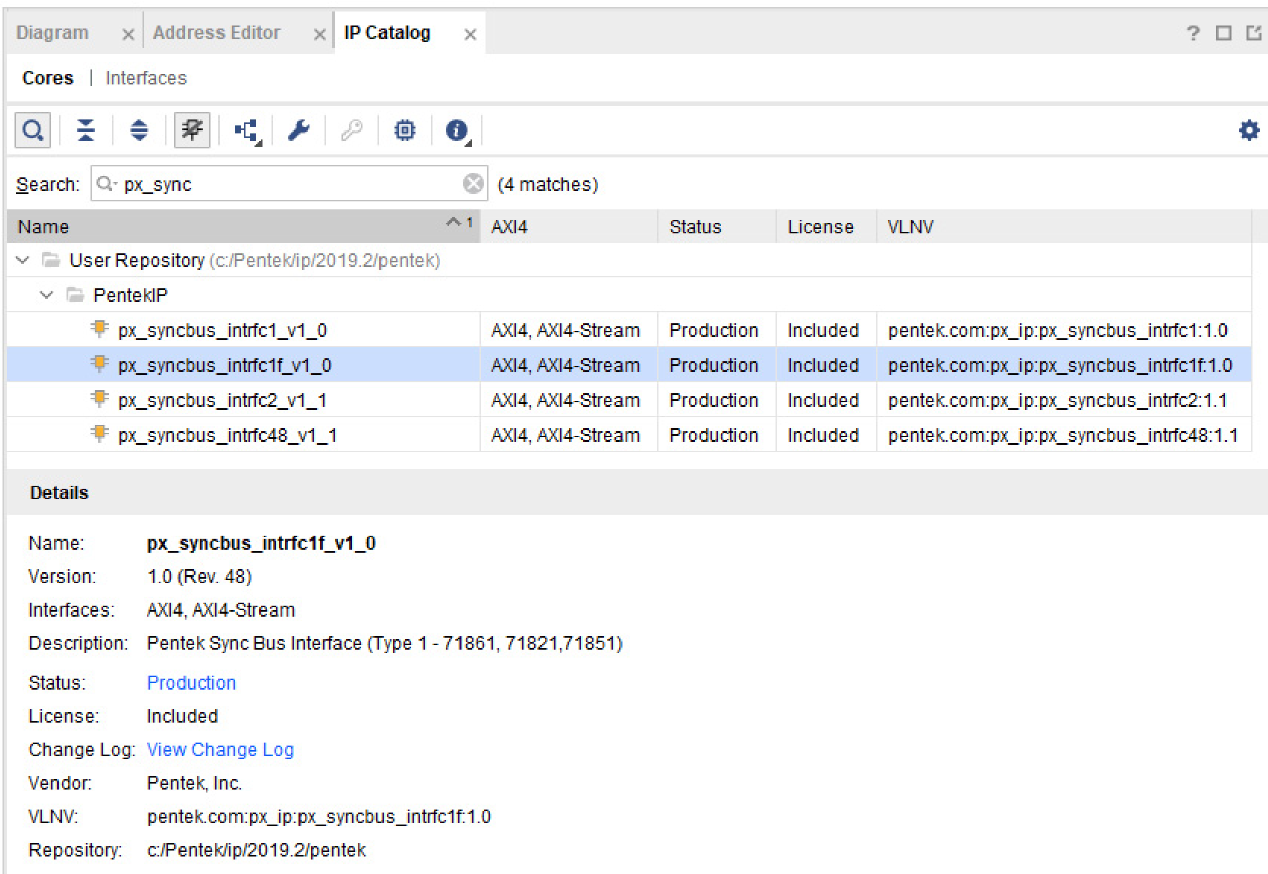
This section is not applicable to this IP core.

Chapter 6: Design Flow Steps

6.1 Pentek IP Catalog

This chapter describes customization and generation of the Pentek Sync Bus Interface 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_syncbus_intrfc1f_v1_0** as shown in Figure 6–1.

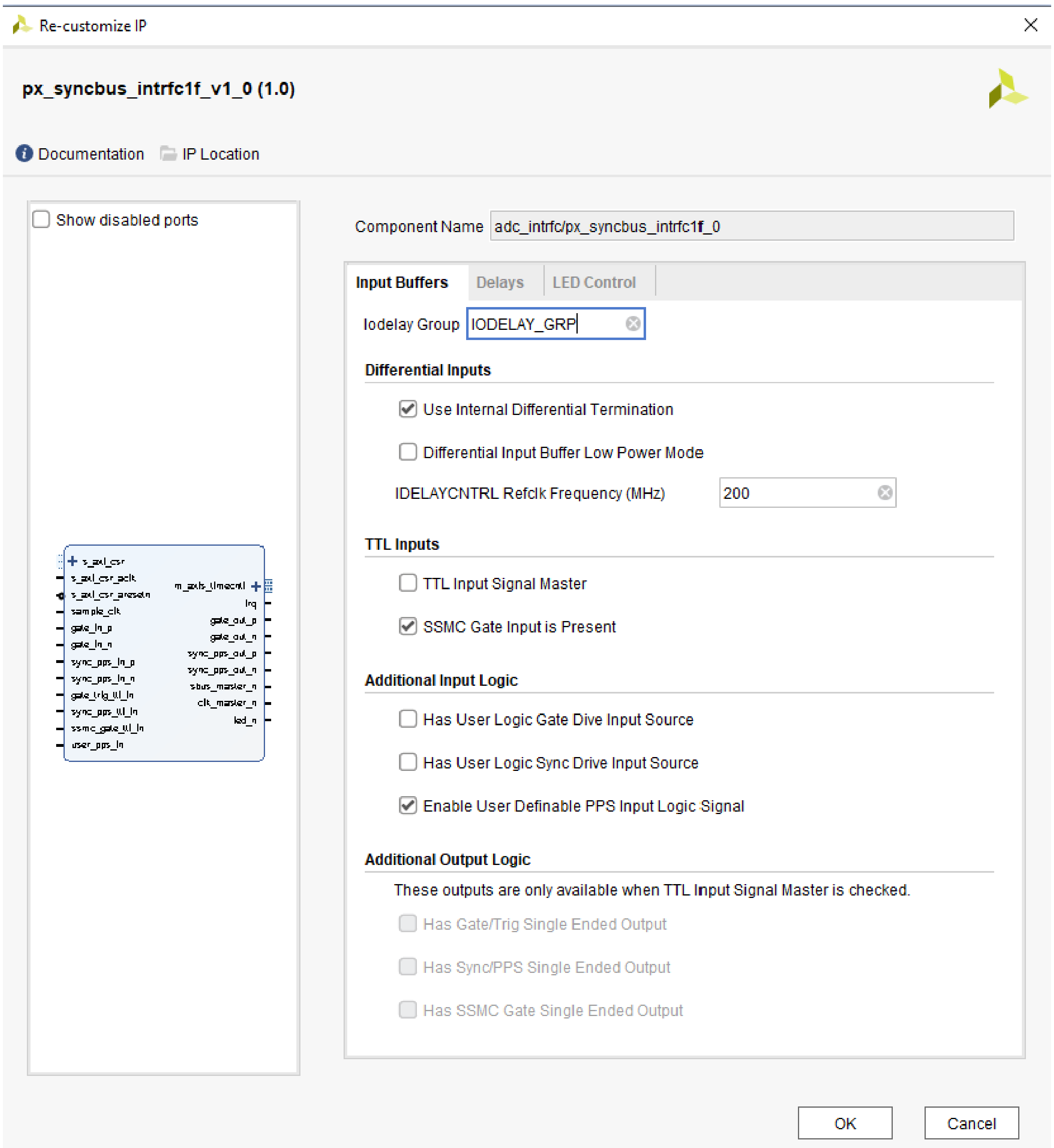
Figure 6–1: Sync Bus Interface Core in Pentek IP Catalog



6.1 Pentek IP Catalog (continued)

When you select the **px_syncbus_intrfc1f_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: Sync Bus Interface Core IP Symbol



6.2 User Parameters

The user parameters of this IP core 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 Sync Bus Interface Core in Vivado Design Suite.

Required Constraints

The XDC constraints are not provided with the Sync Bus Interface Core. Clock con–straints 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 Sample clock (**`sample_clk`**) frequency is not limited.

The AXI4–Lite Interface clock (**`s_axi_csr_aclk`**) has a maximum frequency of 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.

6.5 Simulation

The testbench and the simulation results for this IP core will be available in the next revision of this user manual.

6.6 Synthesis and Implementation

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