LogiCORE IP AXI-Stream FIFO (v1.00a)

DS806 September 21, 2010 **Product Specification**

Introduction

The AXI-Stream FIFO core allows memory mapped access to a AXI-Stream interface. The core can be used to interface to the AXI Ethernet without the complexity or resource utilization of using DMA.

The principal operation of this core allows the write or read of data packets to or from a device without any concern over the AXI4-Stream interface signalling. The management of the AXI4-Stream interfaces is transparent to the user.

Features

- 32-bit AXI4-Lite slave interface
- Independent internal 2 KB TX and RX data FIFOs
- Full duplex operation.
- Supports AXI Ethernet basic mode only
- Provides interrupts for error and status conditions

LogiCORE IP Facts Table					
Core Specifics					
Supported Device Family[1]	Virtex-6, Spartan-6				
Supported User Interfaces	AXI4-Lite , AXI4-Stream				
	Resources				**Frequency**
Configuration	LUTs	FFs	DSP Slices	Block RAMs	Max. Frequency
Configuration 1	See Table 14 and Table 15.				
Provided with Core					
Documentation	Product Specification				
Design Files	VHDL				
Example Design	Not Provided				
Test Bench	Not Provided				
Constraints File	Not Provided				
Simulation Model	N/A				
Tested Design Tools					
Design Entry Tools	12.3 EDK				
Simulation	Mentor Graphics ModelSim 6.5c				
Synthesis Tools	XST				
Support					
Provided by Xilinx, Inc.					

Notes:

1. For a listing of supported devices, see the release notes for this core.
2. For more information on the Spartan-6 devices, see the DS160 Spartan-6 Family Overview.
3. For more information on the Virtex-6 devices, see the DS150 Virtex-6 Family Overview.

Overview

Figure 1 shows the major components in the AXI-Stream FIFO core: an AXI Interface block with an AXI4-Lite Slave interface, Interrupt Controller, a Registers module, a Receive Control Module, a Transmit Control Module, a Receive FIFO for the receive data and length, and a Transmit FIFO for the transmit data and the length.

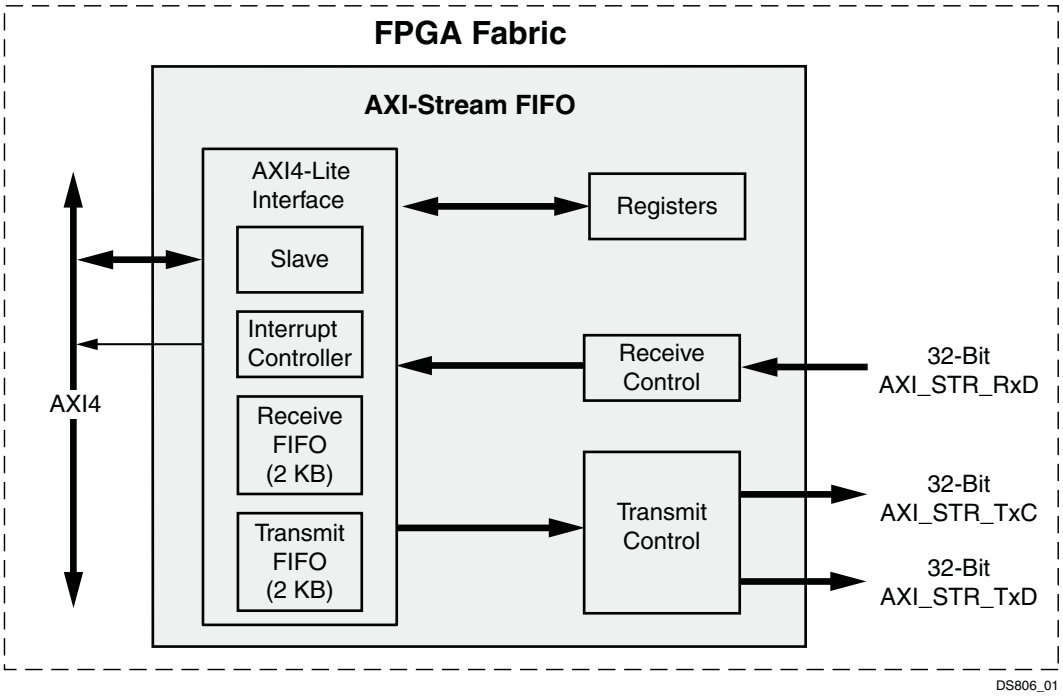

Figure 1: **AXI-Stream FIFO Core Block Diagram**

I/O Signals

The AXI-Stream FIFO core uses a *transparent bus* EDK format to simplify the connection of signals between the AXI4-Stream interface of the core and AXI4-Stream interfaces of other IP such as the AXI Ethernet core. This technique allows the EDK tools to automatically connect the AXI signals as an entire group.

When using AXI-Stream FIFO core with the AXI Ethernet core, connect the three AXI-Stream interfaces below:

1. AXI_STR_TXD - Axi_Stream Transmit Data
2. AXI_STR_TXC - Axi_Stream Transmit Control
3. AXI_STR_RXD - Axi_Stream Receive Data

In addition, to support the transmit protocol of the AXI Ethernet core, add the AXI_STR_TxC interface. This interface does not provide any actual data related to the transmit packet data.

The AXI-Stream FIFO core uses one clock from the AXI4-Lite interface for all clock inputs. When the AXI Ethernet core is used with the AXI-Stream FIFO core, all the AXI Stream input clocks of the AXI Ethernet core must use the same clock.

Table 1: **I/O Signals**

Signal Name	Direction	Description
Top Level System Signals		
Interrupt	O	System Interrupt
AXI4-Lite Signals		
S_AXI_ACLK	I	AXI Clock
S_AXI_ARESETN	I	AXI Reset, active LOW
S_AXI_AWADDR(C_SAXI_ADDR_WIDTH-1:0)	I	AXI Write address.
S_AXI_AWVALID	I	AXI address valid.
S_AXI_AWREADY	O	AXI address ready.
S_AXI_WDATA(C_SAXI_DATA_WIDTH-1:0)	I	Write data,
S_AXI_WSTRB((C_SAXI_DATA_WIDTH/8)-1:0)	I	Write strobes.
S_AXI_WVALID	I	Write valid.
S_AXI_WREADY	O	Write ready.
S_AXI_BRESP	I	Write response.
S_AXI_BREADY	O	Write response ready.
S_AXI_ARADDR(C_SAXI_ADDR_WIDTH-1:0)	I	Read address.
S_AXI_ARVALID	I	Read address valid.
S_AXI_ARREADY	O	Read address ready.
S_AXI_RDATA(C_SAXI_DATA_WIDTH-1:0)	I	Read data.
S_AXI_RRESP(1:0)	I	Read response.
S_AXI_RVALID	I	Read valid.
S_AXI_RREADY	O	Read ready.
AXI-Stream Transmit Data Channel Signals		
AXI_STR_TXD_ACLK	I	Clock for the AXI-Stream Transmit data interface.
AXI_STR_TXD_ARESETN	O	Reset for the AXI-Stream Transmit data interface
AXI_STR_TXD_TVALID	O	Transmit data channel valid
AXI_STR_TXD_TREADY	I	Transmit data channel ready
AXI_STR_TXD_TLAST	O	Transmit data channel last word
AXI_STR_TXD_TSTRB(3:0)	O	Transmit data channel byte strobes
AXI_STR_TXD_TDATA(31:0)	O	Transmit data channel data
AXI-Stream Transmit Control Channel Signals		
AXI_STR_TXC_ACLK	I	Clock for the AXI-Stream Transmit Control interface
AXI_STR_TXC_ARESETN	O	Reset for the AXI-Stream Transmit Control interface
AXI_STR_TXC_TVALID	O	Transmit Control channel valid
AXI_STR_TXC_TREADY	I	Transmit Control channel ready
AXI_STR_TXC_TLAST	O	Transmit Control channel last word
AXI_STR_TXC_TSTRB(3:0)	O	Transmit Control channel byte strobes

Table 1: **I/O Signals *(Cont'd)***

Signal Name	Direction	Description
AXI_STR_TXC_TDATA(31:0)	O	Transmit Control channel control
AXI-Stream Receive Data Channel Signals		
AXI_STR_RXD_ACLK	I	Clock for the AXI-Stream Receive Data interface
AXI_STR_RXD_ARESETN	O	Reset for the AXI-Stream Receive Data interface
AXI_STR_RXD_TVALID	I	Receive Data channel valid
AXI_STR_RXD_TREADY	O	Receive Data channel ready
AXI_STR_RXD_TLAST	I	Receive Data channel last word
AXI_STR_RXD_TSTRB(3:0)	I	Receive Data channel byte strobes
AXI_STR_RXD_TDATA(31:0)	I	Receive Data channel data

Design Parameters

To allow the user to generate a AXI-Stream FIFO core that is tailored for their system, certain features are parameterizable, thereby resulting in a design that utilizes only the resources required by their system and runs at the best possible performance. The features that are parameterizable in the AXI-Stream FIFO core design are shown in Table 2.

Inferred Parameters

In addition to the parameters listed in Table 2, there are parameters that are inferred for each AXI interface in the EDK tools. Through the design, these EDK-inferred parameters control the behavior of the AXI Interconnect. For a complete list of the interconnect settings related to the AXI interface, see the DS768 AXI Interconnect Specification Data Sheet.

Table 2: **Design Parameters**

Feature/Description	Parameter Name	Allowable Values	Default Values	VHDL Type
Device family	C_FAMILY	virtex6, spartan6	spartan6	string
Width of all ID signals across the interface. This is auto computed by the tools	C_SAXI_ID_WIDTH	Integer	4	Integer
Address width of the interface. Must be 32.	C_SAXI_ADDR_WIDTH	32	32	Integer
Data width of the interface. For this core it will always be 32.	C_SAXI_DATA_WIDTH	32	32	Integer
Base address of the IP.(1)	C_SAXI_BASEADDR	0x0 - 0xffffffff	0xffffffff	Bit32
High address of the IP.(1)	C_SAXI_HIGHADDR	0x0 - 0xffffffff	0x00000000	Bit32

Notes:

1. HIGHADDR is required to be at least BASEADDR + 524,287 in order to provide space for the 32-bit addresses used by the registers and memory. For example: C_S_AXI_BASEADDR = 0x00000000 and C_X_AXI_HIGHADDR = 0x0007FFFF. The range specified by BASEADDR and HIGHADDR must be a power of 2 in size, and must have BASEADDR aligned to the size.

Figure 2 shows the AXI-Stream FIFO core connected to the AXI Ethernet core.

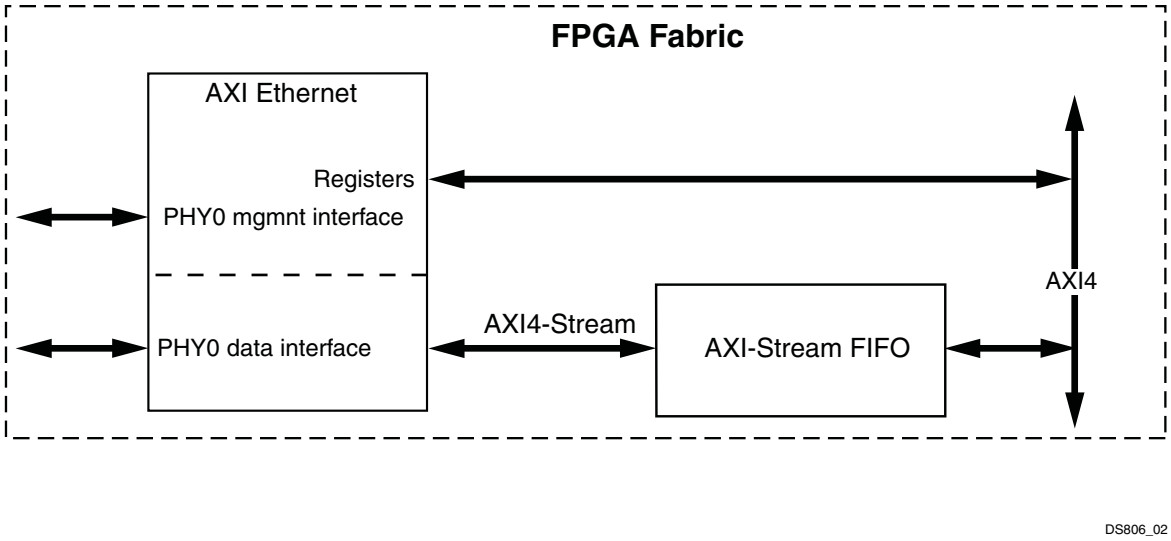

Figure 2: **AXI-Stream FIFO Connected to an AXI Ethernet Core**

Figure 3 shows a partial code segment from an EDK MHS file. The file shows an example connection between two AXI-Stream FIFO cores and an AXI Ethernet core.

```
BEGIN axi_fifo_mm_s
 PARAMETER INSTANCE = ETHERNET_fifo
 PARAMETER HW_VER = 1.00.a
 PARAMETER C_INTERCONNECT_S_AXI_MASTERS = microblaze_0.M_AXI_DP
 PARAMETER C_BASEADDR = 0x71a00000
 PARAMETER C_HIGHADDR = 0x71a0ffff
 BUS_INTERFACE S_AXI =    axi_interconnect_memory_mapped_lite_0
 BUS_INTERFACE AXI_STR_RXD = ETHERNET_fifo_rxd
 BUS_INTERFACE AXI_STR_TXD = ETHERNET_fifo_txd
 BUS_INTERFACE AXI_STR_TXC = ETHERNET_fifo_txc
 PORT S_AXI_ACLK = clk_100_0000MHzMMCM0
 PORT INTERRUPT = axi_fifo_INTERRUPT
 PORT AXI_STR_TXD_ACLK = clk_100_0000MHzMMCM0
 PORT AXI_STR_TXC_ACLK = clk_100_0000MHzMMCM0
 PORT AXI_STR_RXD_ACLK = clk_100_0000MHzMMCM0
END

BEGIN axi_ethernet
…
 BUS_INTERFACE S_AXI = axi_interconnect_memory_mapped_lite_0
 BUS_INTERFACE AXI_STR_RXD = ETHERNET_fifo_rxc
 BUS_INTERFACE AXI_STR_TXD = ETHERNET_fifo_txd
 BUS_INTERFACE AXI_STR_TXC = ETHERNET_fifo_txc
 …
 PORT PHY_RST_N = ETHERNET_PHY_RST_N
 PORT GTX_CLK = clk_125_0000MHz
 PORT INTERRUPT = ETHERNET_INTERRUPT
 PORT AXI_STR_RXS_TREADY = net_vcc
 PORT AXI_STR_TXC_ACLK = clk_100_0000MHzMMCM0
 PORT AXI_STR_TXD_ACLK = clk_100_0000MHzMMCM0
 PORT AXI_STR_RXS_ACLK = clk_100_0000MHzMMCM0
 PORT AXI_STR_RXD_ACLK = clk_100_0000MHzMMCM0
END
```

DS806_03

Figure 3: **EDK MHS File Code Segment**

Registers Definition

The AXI-Stream FIFO core contains the registers listed in Table 3.

Table 3: **Register Names and Descriptions**

Register Name	AXI Address	Access
Interrupt Status Register (ISR)	C_SAXI_BASEADDR + 0x0	Read/Clear on Write(1)
Interrupt Enable Register (IER))	C_SAXI_BASEADDR + 0x4	Read/Write
Transmit data FIFO reset (TDFR)	C_SAXI_BASEADDR + 0x8	Write(2)
Transmit data FIFO Vacancy (TDFV)	C_SAXI_BASEADDR + 0xC	Read
Transmit data FIFO 32bit wide data write port (TDFD)	C_SAXI_BASEADDR + 0x10	Write
Transmit Length FIFO (TLF)	C_SAXI_BASEADDR + 0x14	Write
Receive data FIFO reset (RDFR)	C_SAXI_BASEADDR + 0x18	Write(2)
Receive data FIFO Occupancy (RDFO)	C_SAXI_BASEADDR + 0x1C	Read
Receive data FIFO 32bit wide data read port (RDFD)	C_SAXI_BASEADDR + 0x20	Read
Receive Length FIFO (RLF)	C_SAXI_BASEADDR + 0x24	Read
AXI-Stream reset (SRR)	C_SAXI_BASEADDR + 0x28	Write(2)
Reserved	C_SAXI_BASEADDR + 0x2C to C_SAXI_BASEADDR + 0x3C	N/A(3)

Notes:

1. The latched interruptible condition is cleared by writing a 1 to that bit location. Writing a 1 to a bit location that is 0 has no effect. Likewise, writing a 0 to a bit location that is 1 has no effect. Multiple bits may be cleared in a single write.
2. Reset if written with 0xA5.
3. If read, these registers will return 0x0. Writing these registers will have no effect.

Interrupt Interface

The interrupt signals generated by the AXI-Stream FIFO core are managed by the ISR and IER registers. The ISR is combined with the IER register to define the interrupt interface of the AXI-Stream FIFO core. An overview diagram of the interrupt control structure is show in Figure 4.

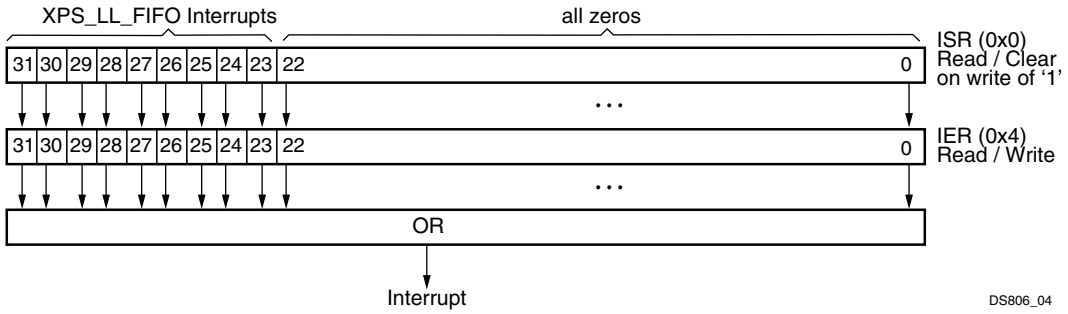

Figure 4: **Interrupt Control Structure**

Interrupt Status Register (ISR)

The Interrupt Status Register is shown in Figure 5. The Interrupt Status register uses one bit to represent each internal interruptible condition.

Once an interruptible condition occurs, it will be captured in this register (represented as the corresponding bit being set to 1) even if the condition changes. The latched interruptible condition is cleared by writing a 1 to its bit location. Writing a 1 to a bit location that is 0 has no effect. Likewise, writing a 0 to a bit location that is 1 has no effect. Multiple bits may be cleared in a single write.

For any bit set in the Interrupt Status Register, a corresponding bit must be also set in the Interrupt Enable Register for the IP2INTC_Irpt signal to be driven active High out of the AXI-Stream FIFO core.

The Interrupt Status Register bit definitions are detailed in Table 4.

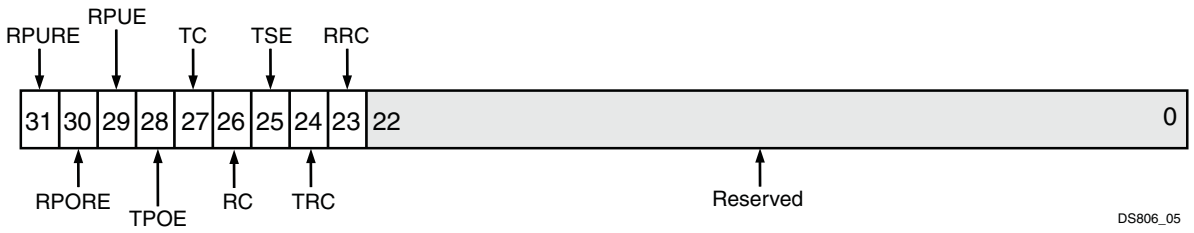

Figure 5: **Interrupt Status Register (offset 0x0)**

Table 4: **Interrupt Status Register Bit Definitions**

Bit(s)	Name	Core Access	Reset Value	Description
0-22	Reserved	Read	0x0	Reserved: These bits are reserved for future definition and will always return all zeros.
23	RRC	Read/Clear on Write of "1"	0	**Receive Reset Complete:** This interrupt indicates that a reset of the receive logic has completed. "0" = No interrupt pending. "1" = Interrupt pending.
24	TRC	Read/Clear on Write of "1"	0	**Transmit Reset Complete:** This interrupt indicates that a reset of the transmit logic has completed. "0" = No interrupt pending. "1" = Interrupt pending.
25	TSE	Read/Clear on Write of "1"	0	**Transmit Size Error:** This interrupt is generated if the number of 32-bit words written to the transmit data FIFO does not match the value written to the transmit length register (bytes) divided by 4. A reset of the transmit logic is required to recover. "0" = No interrupt pending. "1" = Interrupt pending.
26	RC	Read/Clear on Write of "1"	0	**Receive Complete:** Indicates that at least one successful receive has completed and that the receive packet data and packet data length is available. This signal is not set for unsuccessful receives. This interrupt may represent more than one packet received, so it is important to check the receive data FIFO occupancy value to determine if additional receive packets are ready to be processed. "0" = No interrupt pending. "1" = Interrupt pending.
27	TC	Read/Clear on Write of "1"	0	**Transmit Complete:** Indicates that at least one transmit has completed. "0" = No interrupt pending. "1" = Interrupt pending.

Table 4: **Interrupt Status Register Bit Definitions *(Cont'd)***

Bit(s)	Name	Core Access	Reset Value	Description
28	TPOE	Read/Clear on Write of "1"	0	**Transmit Packet Overrun Error:** This interrupt is generated if an attempt is made to write to the transmit data FIFO when it is full. A reset of the transmit logic is required to recover. "0" = No interrupt pending. "1" = Interrupt pending.
29	RPUE	Read/Clear on Write of "1"	0	**Receive Packet Underrun Error:** This interrupt occurs when an attempt is made to read the receive FIFO when it is empty. The data read is not valid. A reset of the receive logic is required to recover. "0" = No interrupt pending. "1" = Interrupt pending.
30	RPORE	Read/Clear on Write of "1"	0	**Receive Packet Overrun Read Error**: This interrupt occurs when more words are read from the receive data FIFO than are in the packet being processed. Even though the FIFO may not be empty, the read has gone beyond the current packet and removed data from the next packet. A reset of the receive logic is required to recover. "0" = No interrupt pending. "1" = Interrupt pending.
31	RPURE	Read/Clear on Write of "1"	0	**Receive Packet Underrun Read Error**: This interrupt occurs when an attempt is made to read the receive length register when it is empty. The data read is not valid. A reset of the receive logic is required to recover. "0" = No interrupt pending. "1" = Interrupt pending.

Interrupt Enable Register (IER)

The Interrupt Enable Register shown in Figure 6 determines which interrupt sources in the Interrupt Status Register are allowed to generate interrupts. Setting to "1" in a bit location enables the related interrupt from being propagated, while a value of "0" disables it.

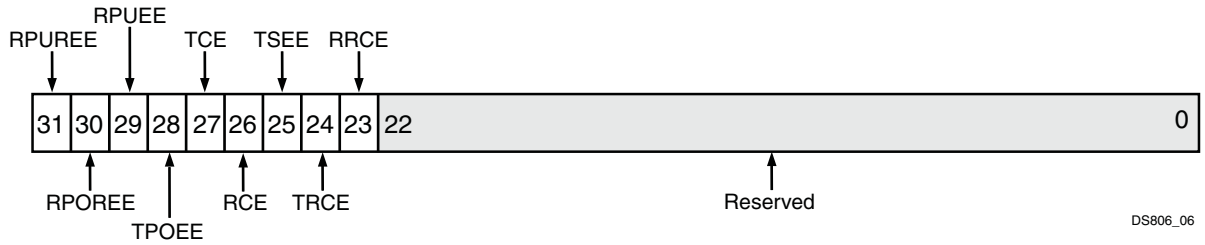

Figure 6: **Interrupt Enable Register (offset 0x4)**

Transmit Data FIFO Reset Register (TDFR)

The Transmit Data FIFO Reset Register shown in Figure 7 is not an actual register, but is instead a write-only address, which when written with a specific value, generates a reset for the Transmit Data FIFO. This reset will not occur until transmit activity on the TX AXI Stream has completed. The reset can occur only during inactive times on the TX AXI Stream and will affect only the transmit circuitry in this core, thereby preventing the core on the other end of the AXI-Stream from receiving a partial packet which could potentially cause a failure condition in the latter core.

Because of this mode of operation, it is possible that if the AXI-Stream becomes unresponsive in the middle of a AXI-Stream transaction, a reset will never occur. For example, this might occure while waiting for the destination ready to go active in the middle of a transfer. In such cases it is necessary to use both the AXI-Stream Reset and the S_AXI_ARESETN reset.

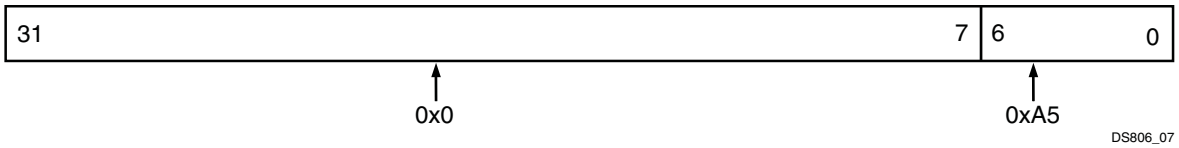

Figure 7: **Transmit Data FIFO Reset Register (offset 0x8)**

Table 5: **Transmit Data FIFO Reset Register Bit Definitions**

Bit(s)	Name	Core Access	Reset Value	Description
31-0	Reset Key	Write	N/A	Reset Write Value. "0x000000A5" - Generate a reset. Others - No effect.

Transmit Data FIFO Vacancy Register (TDFV)

The Transmit Data FIFO Vacancy Register shown in Figure 8 is a read-only register that gives the vacancy status of the Transmit Data FIFO.

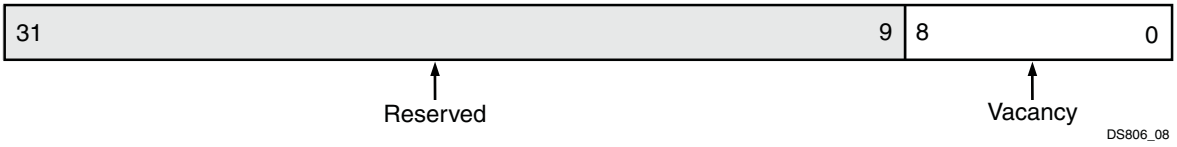

Figure 8: **Transmit Data FIFO Vacancy Register (offset 0xC)**

Transmit Data FIFO Data Write Port (TDFD)

The Transmit Data FIFO Data Write Port shown in Figure 9 is a 32-bit wide address location for writing data into the Transmit Data FIFO. The smallest packet that may be transmitted is four 32-bit words which corresponds to 13 to 16 bytes. The maximum packet that may be transmitted is limited by the size of the FIFO which is 510 words or 2037 to 2040 bytes.

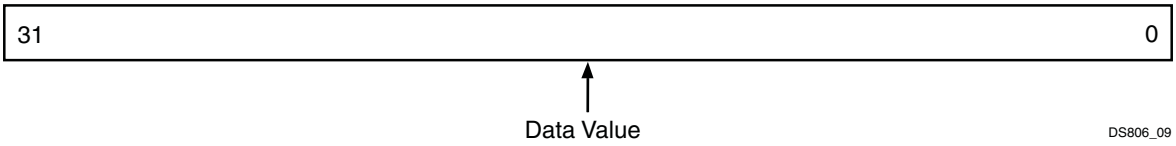

Figure 9: **Transmit Data FIFO Data Write Port (offset 0x10)**

Table 6: **Transmit Data FIFO Data Write Port Bit Definitions**

Bit(s)	Name	Core Access	Reset Value	Description
31-0	Write Data Value	Write	N/A	Transmit Data FIFO Write Value.

Receive Data FIFO Reset Register (RDFR)

The Receive Data FIFO Reset Register shown in Figure 10 is not an actual register but, rather a write-only address, which when written with a specific value, generates a reset for the Receive Data FIFO.

This reset will not occur until receive activity on the RX AXI Stream has completed. Only during inactive times on the RX AXI Stream can a reset occur. It will affect only the receive circuitry in this core. This prevents the core on the other end of the AXI-Stream from transmitting a partial packet which may cause failure condition in that core.

Because of this mode of operation, it is possible that if the AXI-Stream interface becomes unresponsive during an AXI-Stream transaction, that the reset will never occur. An example transaction is if a packet is received over the AXI-Stream that exceeds the FIFO size of this core causing the core's destination ready to become inactive in the middle of a transfer. In this case, an S_AXI_ARESETN reset is needed.

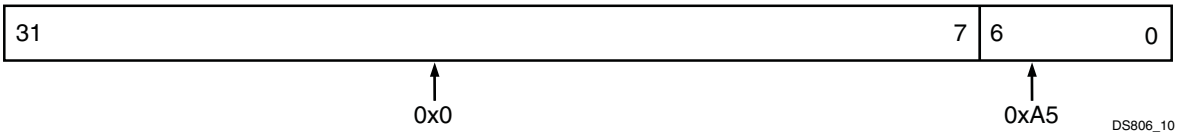

Figure 10: **Receive Data FIFO Reset Register (offset 0x18)**

Table 7: **Receive Data FIFO Reset Register Bit Definitions**

Bit(s)	Name	Core Access	Reset Value	Description
31-0	Reset Key	Write	N/A	Reset Write Value: "0x000000A5" - Generate a reset. Others - No effect.

Receive Data FIFO Occupancy Register (RDFO)

The Receive Data FIFO Occupancy Register shown in Figure 11 is a read-only register that gives the occupancy status of the Receive Data FIFO.

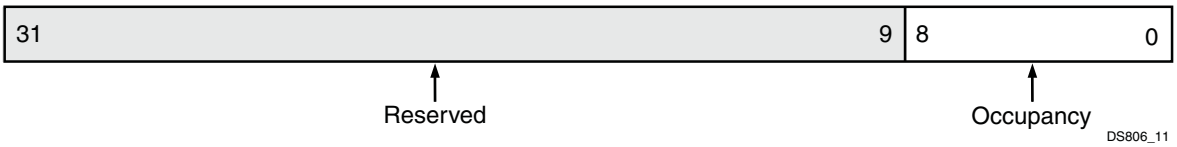

Figure 11: **Receive Data FIFO Occupancy Register (offset 0x1C)**

Table 8: **Receive Data FIFO Occupancy Register Bit Definitions**

Bit(s)	Name	Core Access	Reset Value	Description
8-0	Occupancy	Read	0x0	Receive Data FIFO Occupancy: This is the unsigned value reflecting a current snapshot of the number of 32-bit wide locations in use for data storage in the receive Data FIFO memory core. This value is only updated after a packet is successfully received, and therefore can be used to determine if a receive packet is ready to be processed when a non-0 value is read.
31-9	Reserved	Read	0x0	Reserved: These bits are reserved for future definition and will always return all zeros.

Receive Data FIFO Data Read Port (RDFD)

The Receive Data FIFO Data Read Port shown in Figure 12 is a 32-bit wide address location for reading data from the Receive Data FIFO. The smallest packet that may be received is a four 32-bit words packet which corresponds to 13 to 16 bytes. The maximum packet that may be received is limited by the size of the FIFO which is 510 words or 2037 to 2040 bytes.

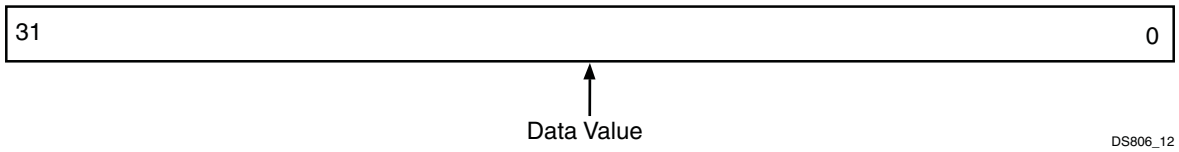

Figure 12: **Receive Data FIFO Data Read Port (offset 0x20)**

Table 9: Receive Data FIFO Data Read Port Bit Definitions

Bit(s)	Name	Core Access	Reset Value	Description
31-0	Read Data Value	Read	N/A	Receive Data FIFO Read Value.

Transmit Length Register (TLR)

The Transmit Length Register shown in Figure 13 is used to store packet length values (the number of bytes in the packet) corresponding to valid packets ready for transmit. The data for the packet is stored in the transmit Data FIFO. The data is written to the AXI-Stream FIFO core over the AXI4 interface, typically by a processor or DMA core such the Central DMA (CDMA). When presenting a transmit packet to the AXI-Stream FIFO core, write the packet data to the Transmit Data FIFO first, then write the length of the packet into the TLR.

Once the packet length is written to the TLR, it is automatically moved to the Transmit Data FIFO with the packet data freeing up the TLR for another value. The packet length must be written to the TLR after the packet data is written to the transmit data FIFO. It is **not** valid to write data for multiple packets to the transmit data FIFO before writing the packet length values.

The action of writing to the Transmit Length Register is used by the AXI-Stream FIFO core to initiate the processing of transmit packets across the AXI-Stream interface. This action continues until all of the TLR values that have been stored are processed.

The width of the TLR is wide enough to support packets up to 2048 bytes in length. The smallest packet that may be transmitted is four 32-bit words which corresponds to 13 to 16 bytes. The maximum packet that may be transmitted is limited by the size of the FIFO which is 510 words or 2037 to 2040 bytes.

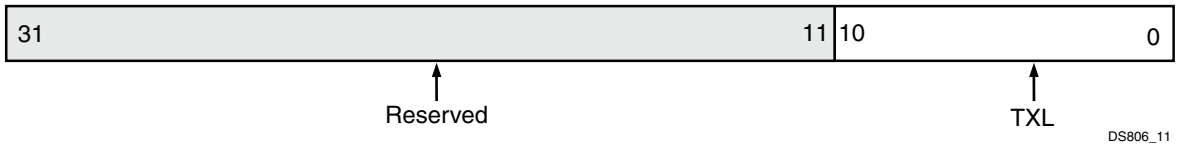

Figure 13: **Transmit Length Register (offset 0x14)**

Table 10: **Transmit Length Register Bit Definitions**

Bit(s)	Name	Core Access	Reset Value	Description
31-11	Reserved	N/A	0x0	Reserved**:** These bits are reserved for future definition and will always return all zeros.
10-0	TXL	Write	0x0	Transmit Length: The number of bytes of the corresponding transmit packet stored in the transmit data FIFO.

Receive Length Register (RLR)

The receive length register shown in Figure 14 is used to retrieve packet length values (the number of bytes in the packet) corresponding to valid packets received. The data for the packet is stored in theReceive Data FIFO.

The length is written by the AXI-Stream FIFO core when the packet is received across the RX AXI-Stream interface and the receive data FIFO has the adequate number of locations to ensure that all of the packet data has been saved.

The RLR should only be read when a receive packet is available for processing (the receive occupancy is not zero). Once the RLR is read, the receive packet data should be read from the receive data FIFO before the RLR is read again.

The RLR values are stored in the receive data FIFO by the AXI-Stream FIFO core with the data of each packet. The RLR value for the subsequent packet to be processed is moved to the RLR when the previous RLR value has been read.

This register is wide enough to support packets up to 2048 bytes in length. The smallest packet that may be received is four 32-bit words which corresponds to 13 to 16 bytes. The maximum packet that may be received is limited by the size of the FIFO and is 510 words or 2037 to 2040 bytes.

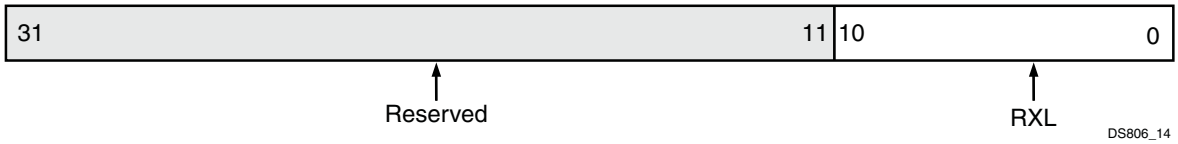

Figure 14: **Receive Length Register (offset 0x24)**

Table 11: Receive Length Register Bit Definitions

Bit(s)	Name	Core Access	Reset Value	Description
31-11	Reserved	Read	0x0	Reserved: These bits are reserved for future definition and will always return all zeros.
10-0	RXL	Read	0x0	Receive Length: The number of bytes of the corresponding receive data stored in the receive data FIFO.

AXI-Stream Reset Register (SRR)

The AXI-Stream Register shown in Figure 15 is not an actual register. It is a write-only address, which when written with a specific value, generates an immediate reset for the entire core as well as driving a reset on the external outputs, AXI_STR_RXD_ARESETN, AXI_STR_TXD_ARESETN, and AXI_STR_TXC_ARESETN, which can be used to reset the core on the other end of the AXI-Stream.

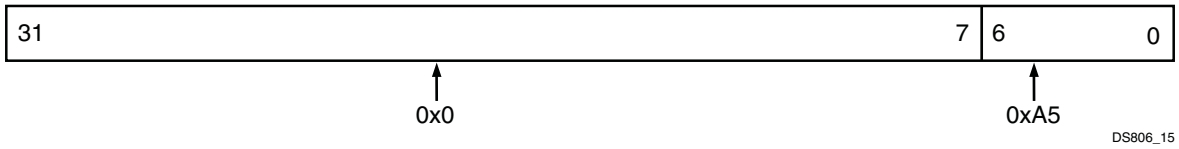

Figure 15: **AXI-Stream Reset Register (offset 0x28)**

Table 12: AXI-Stream Reset Register Bit Definitions

Bit(s)	Name	Core Access	Reset Value	Description
31-0	Reset Key	Write	N/A	Reset Write Value: "0x000000A5" - Generate a reset. Others - No effect.

Reserved Registers

Reading from reserved registers will return zeros and writing to reserved registers will have no effect. However, any accesses to address offset 0x40 and above causes undefined results.

Basic Usage

The AXI-Stream FIFO core was designed to provide memory-mapped access to an AXI-Stream interface connected to other IP, such as the AXI Ethernet core. Systems must be built through the Embedded Development Kit (EDK) to attach the AXI-Stream FIFO core, AXI Ethernet core, processor, memory, interconnect the buses, clocking, and additional embedded components.

This section briefly describes the operation of the AXI-Stream FIFO core through register accesses using the AXI Ethernet core as an example.

Packets will automatically be transmitted by writing packet data to the transmit data FIFO followed by writing a length in the TLR.

Receiving packets functions in a manner similar to transmission, except that the steps are reversed. Receiving the ISR receive complete interrupt or polling of the receive data FIFO occupancy register (when not zero) indicates the reception of a packet. Reading the RLR provides the packet length in bytes. Given the number of received packet bytes, the appropriate number of reads of the Read data FIFO will provide the packet data.

Table 13 illustrates a power-up read of the registers followed by a transmission and reception of a single packet. See the register definitions for further information and options.

Table 13: **Sample TX and RX Usage**

Register	Access	Value	Activity
Power-up Read of Register Values			
ISR	Read Word	0x01800000	Read interrupt status register
ISR	Write Word	0x0FFFFFFFF	Write to clear reset done interrupt bits
ISR	Read Word	0x00000000	Read interrupt status register
IER	Read Word	0x00000000	Read interrupt enable register
TDFV	Read Word	0x000001FE	Read the transmit FIFO vacancy
RDFO	Read Word	0x00000000	Read the receive FIFO occupancy
Transmit a Packet			
TXFIFO_DATA	Write Word	0xFFFFFFFF	4 bytes of data
TXFIFO_DATA	Write Word	0x12345678	4 bytes of data
TXFIFO_DATA	Write Word	0x00010203	4 bytes of data
TXFIFO_DATA	Write Word	0x08090A0B	4 bytes of data
TXFIFO_DATA	Write Word	0x10111213	4 bytes of data
TXFIFO_DATA	Write Word	0x18191A1B	4 bytes of data
TXFIFO_DATA	Write Double	0x20212223	4 bytes of data
TXFIFO_DATA	Write Word	0x28292A2B	4 bytes of data
TDFV	Read Word	0x000001F7	Read the transmit FIFO vacancy
TLR	Write Word	0x00000020	Transmit length (0x20 = 32bytes), this starts transmission
ISR	Read Word	0x08000000	A typical value after Tx Complete is indicated by interrupt
ISR	Write Word	0x0FFFFFFFF	Write to clear reset done interrupt bits
ISR	Read Word	0x00000000	Read interrupt status register
TDFV	Read Word	0x000001FE	Read the transmit FIFO vacancy
Receive a Packet			
ISR	Read Word	0x04000000	A typical value after Rx Complete is indicated by interrupt
ISR	Write Word	0x0FFFFFFFF	Write to clear reset done interrupt bits
ISR	Read Word	0x00000000	Read interrupt status register
RDFO	Read Word	0x00000008	Read the receive FIFO occupancy
RLR	Read Word	0x00000020	Receive length (0x20 =32 bytes) indicates number of bytes to read
RDFO	Read Word	0x00000008	Read the receive FIFO occupancy
RXFIFO_DATA	Read Word	0xFFFFFFFF	4 bytes of data
RXFIFO_DATA	Read Word	0x12345678	4 bytes of data
RXFIFO_DATA	Read Word	0x00010203	4 bytes of data
RXFIFO_DATA	Read Word	0x08090A0B	4 bytes of data
RXFIFO_DATA	Read Word	0x10111213	4 bytes of data
RXFIFO_DATA	Read Word	0x18191A1B	4 bytes of data
RXFIFO_DATA	Read Word	0x20212223	4 bytes of data
RXFIFO_DATA	Read Word	0x28292A2B	4 bytes of packet data and CRC value
RDFO	Read Word	0x00000000	Read the receive FIFO occupancy (no further receive packets to process)

Design Implementation

Design Tools

The AXI-Stream FIFO core design is implemented using VHDL code. Xilinx XST is the synthesis tool used for synthesizing the core.

Target Technology

The target technology is a Virtex®-6 or Spartan®-6 FPGA.

Device Utilization and Performance Benchmarks

Because the AXI-Stream FIFO core will be used with other design pieces in the FPGA, the utilization and timing numbers reported in this section are just estimates. As the AXI-Stream FIFO core is combined with other pieces of the FPGA design, the utilization of FPGA resources and timing of the AXI-Stream FIFO core design will vary from the results reported here. The core benchmarks are shown in Table 14 for a Virtex-6 FPGA and in Table 15 for the Spartan-6 FPGA.

Table 14: **Performance and Resource Utilization Benchmarks for the Virtex-6 FPGA**

Device Resources				F_{MAX} (MHz)
Block RAMs	Slices	Slice Flip- Flops	LUTs	F_{MAX}
2	452	957	876	150

Table 15: **Performance and Resource Utilization Benchmarks for the Spartan-6 FPGA**

Device Resources				F_{MAX} (MHz)
Block RAMs	Slices	Slice Flip- Flops	LUTs	F_{MAX}
2	452	957	876	100

Support

Xilinx provides technical support for this LogiCORE product when used as described in the product documentation. Xilinx cannot guarantee timing, functionality, or support of product if implemented in devices that are not defined in the documentation, if customized beyond that allowed in the product documentation, or if changes are made to any section of the design labeled *DO NOT MODIFY*.

Reference Documents

1. DS562 AXI Slave Burst
2. DS160 Spartan-6 Family Overview
3. DS150 Virtex-6 Family Overview
4. AXI4 AMBA Protocol Version 2.0 Specification

Support

Xilinx provides technical support for this LogiCORE product when used as described in the product documentation. Xilinx cannot guarantee timing, functionality, or support of product if implemented in devices that are not defined in the documentation, if customized beyond that allowed in the product documentation, or if changes are made to any section of the design labeled *DO NOT MODIFY*.

Ordering Information

This Xilinx LogiCORE IP module is provided at no additional cost with the Xilinx ISE Design Suite Embedded Edition software under the terms of the Xilinx End User License. The core is generated using the Xilinx ISE Embedded Edition software (EDK).

Information about this and other Xilinx LogiCORE IP modules is available at the Xilinx Intellectual Property page. For information on pricing and availability of other Xilinx LogiCORE modules and software, please contact your local Xilinx sales representative.

Revision History

Date	Version	Description of Revisions
9/21/10	1.0	Xilinx initial release.

Notice of Disclaimer