AXI4-Stream Switch Core User Guide

UG-CORE-AXI4-STREAM-SWITCH-v1.1
February 2023
www.elitestek.com

Contents

Introduction

The AXI4-Stream Switch core manages traffic on the AXI4-Stream interfaces where it allows you to connect one or more AXI masters to one or more AXI slaves. The AXI4-Stream Switch core uses the AXI4-Stream's `TDEST` signal to route to route transfer to different slaves.

Use the IP Manager to select IP, customize it, and generate files. The AXI4-Stream Switch core has an interactive wizard to help you set parameters. The wizard also has options to create a testbench and/or example design targeting an 易灵思® development board.

Features

- AXI4-Stream interface
- Up to 8 master and 8 slave interfaces
- Supports 2 arbitration modes:
  - Fixed priority
  - Round robin 1
- Arbitration based on:
  - TLAST
  - Maximum number of transfers
  - Number of continuous low TVALID cycles
- Verilog HDL RTL and simulation testbench

FPGA Support

The AXI4-Stream Switch core supports all Trion® and 钛金系列 FPGAs.

Resource Utilization and Performance

Note: The resources and performance values provided are just guidance and change depending on the device resource utilization, design congestion, and user design.

钛金系列 Resource Utilization and Performance

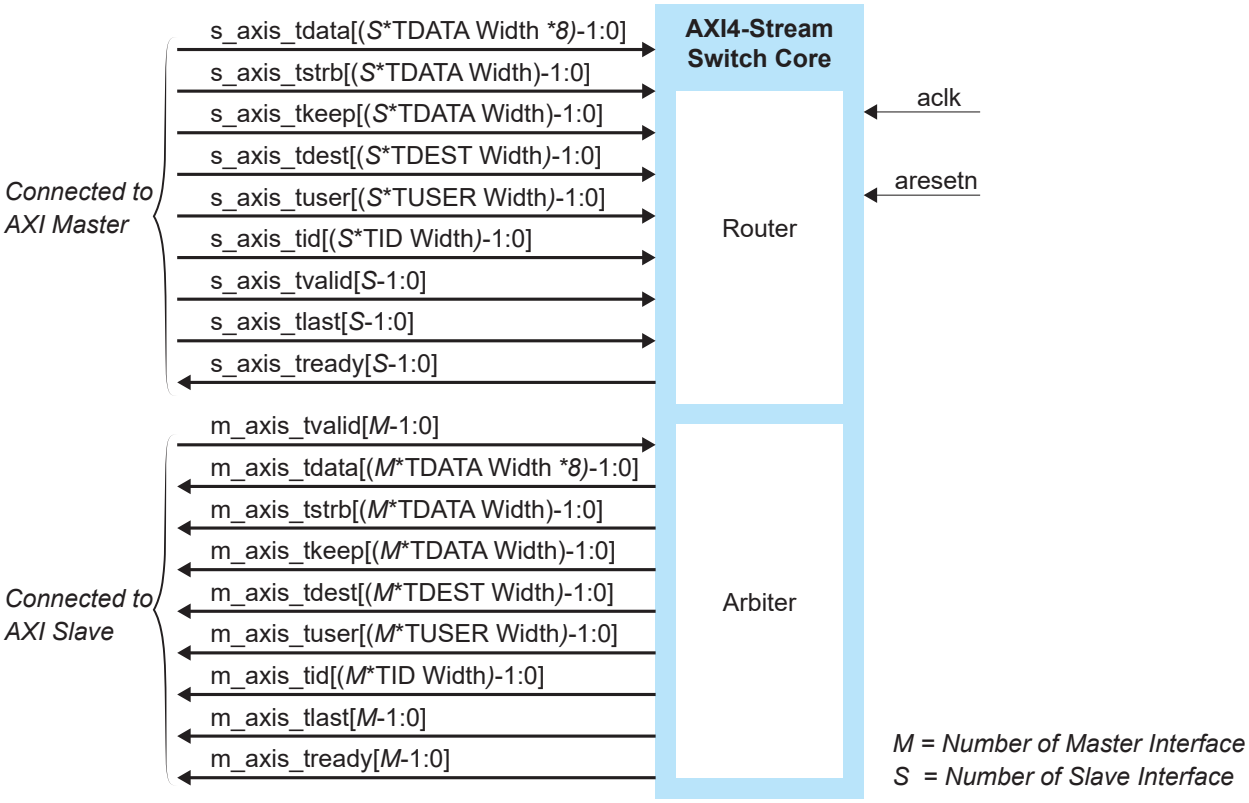
FPGA	Configuration	Logic and Adders	Flip-flops	Memory Blocks	DSP Blocks	f_{MAX} (MHz)	Efinity® Version
Ti60 F225 C4	4-to-2 with single-byte data width	141	66	0	0	433	2022.1

(1) Using Verilog HDL.

Functional Description

The AXI4-Stream Switch core consists of the following blocks:

Figure 1: AXI4-Stream Switch System Block Diagram



Each of the AXI4-stream ports carries the routing information through the TDEST (`s_axis_tdest`) signal. The router decodes the transaction destination. If there is more than a transaction is targeting the same destination simultaneously, the arbiter grants the master based on the selected arbitration scheme. The IP acknowledges the granted port by asserting the TREADY (`s_axis_tready`) of the granted port. The TREADY of other masters remains low. The granted master can now initiate the transaction.

There are multiple options for you to arbitrate the data transaction. You can set the IP to arbitrate based on:

- *TLAST signal*—The master asserts the TLAST (`s_axis_tlast`) signal to indicate the end of a transfer
- *Maximum number of transfers*—User defined number of transfers for each master
- *Number of LOW TVALID Cycles*—User defined timeout counts for each master transaction

Ports

Table 1: Global

Port	Signal Direction	Description
aclk	Input	Core clock.
aresetn	Input	Active low asynchronous reset.

Table 2: Slave Interface (Connected to Master)

S is the number of slave interfaces.

Port	Direction	Description[(2)]
s_axis_tdata [(*S**TDATA Width*8)-1:0]	Input	TDATA is the primary payload that is used to provide the data that is passing across the interface. The width of the data payload is an integer number of bytes.
s_axis_tstrb [(*S**TDATA Width)-1:0]	Input	TSTRB is the byte qualifier that indicates whether the content of the associated byte of TDATA is processed as a data byte or a position byte.
s_axis_tkeep [(*S**TDATA Width)-1:0]	Input	TKEEP is the byte qualifier that indicates whether the content of the associated byte of TDATA is processed as part of the data stream. Associated bytes that have the TKEEP byte qualifier deasserted are null bytes and can be removed from the data stream.
s_axis_tdest [(*S**TDEST Width)-1:0]	Input	TDEST provides routing information for the data stream.
s_axis_tuser [(*S**TUSER Width)-1:0]	Input	TUSER is user-defined sideband information that can be transmitted alongside the data stream.
s_axis_tid [(*S**TID Width)-1:0]	Input	TID is the data stream identifier that indicates different streams of data.
s_axis_tvalid [*S*-1:0]	Input	TVALID indicates the master is driving a valid transfer. A transfer occurs when both TVALID and TREADY are asserted.
s_axis_tlast [*S*-1:0]	Input	TLAST indicates the boundary of a packet.
s_axis_tready [*S*-1:0]	Output	TREADY indicates that the slave can accept a transfer in the current cycle.

(2) Port descriptions extracted from AMBA 4 AXI4-Stream Protocol Specifications.

Table 3: Master Interface (Connected to Slave)

M is the number of master interfaces.

Port	Direction	Description[2]
m_axis_tvalid [*M*-1:0]	Output	TVALID indicates that the master is driving a valid transfer. A transfer takes place when both TVALID and TREADY are asserted.
m_axis_tdata [*M**(TDATA Width*8)-1:0]	Output	TDATA is the primary payload that is used to provide the data that is passing across the interface. The width of the data payload is an integer number of bytes.
m_axis_tstrb [(*M**TDATA Width)-1:0]	Output	TSTRB is the byte qualifier that indicates whether the content of the associated byte of TDATA is processed as a data byte or a position byte.
m_axis_tkeep [(*M**TDATA Width)-1:0]	Output	TKEEP is the byte qualifier that indicates whether the content of the associated byte of TDATA is processed as part of the data stream. Associated bytes that have the TKEEP byte qualifier deasserted are null bytes and can be removed from the data stream.
m_axis_tdest [(*M**TDEST Width)-1:0]	Output	TDEST provides routing information for the data stream.
m_axis_tuser [(*M**TUSER Width)-1:0]	Output	TUSER is user-defined sideband information that can be transmitted alongside the data stream.
m_axis_tid [(*M**TID Width)-1:0]	Output	TID is the data stream identifier that indicates different streams of data.
m_axis_tlast [*M*-1:0]	Output	TLAST indicates the boundary of a packet.
m_axis_tready [*M*-1:0]	Input	TREADY indicates that the slave can accept a transfer in the current cycle.

Arbitration Modes

The AXI4-Stream Switch core includes arbiter engines that grant the request to an AXI master when more than one AXI master issues a request. The AXI4-Stream Switch supports three types of arbitration modes.

Fixed Priority

In this mode, the arbiter always prioritizes the most significant bit (MSB) ports as indicated by the bits shown in bold in the following example. Lower priority ports suffer from starvation.

Request	11100000	11100000	11100000	1110010	00001011	00000000	00001111	11111111
Grant	**1**0000000	**1**0000000	**1**0000000	**1**0000000	0000**1**000	00000000	0000**1**000	**1**0000000

Round Robin 1

The arbitration starts from the MSB port. When more than two transaction requests are issued at the same time, the arbiter grants the request to the port sitting on the right-hand side nearest to the previously served port.

Example:

Request	11100000	11100000	11100000	1110010	00001011	00000000	00001111	11111111
Grant	**1**0000000	0**1**000000	00**1**00000	0000000**1**0	0000000**1**	00000000	0000**1**000	0000**1**00

IP Manager

The Efinity® IP Manager is an interactive wizard that helps you customize and generate 易灵思® IP cores. The IP Manager performs validation checks on the parameters you set to ensure that your selections are valid. When you generate the IP core, you can optionally generate an example design targeting an 易灵思 development board and/or a testbench. This wizard is helpful in situations in which you use several IP cores, multiple instances of an IP core with different parameters, or the same IP core for different projects.

Note: Not all 易灵思 IP cores include an example design or a testbench.

Generating a Core with the IP Manager

The following steps explain how to customize an IP core with the IP Configuration wizard.

1. Open the IP Catalog.
2. Choose an IP core and click **Next**. The **IP Configuration** wizard opens.
3. Enter the module name in the **Module Name** box.

   Note: You cannot generate the core without a module name.
4. Customize the IP core using the options shown in the wizard. For detailed information on the options, refer to the IP core's user guide or on-line help.
5. (Optional) In the **Deliverables** tab, specify whether to generate an IP core example design targeting an 易灵思® development board and/or testbench. For SoCs, you can also optionally generate embedded software example code. These options are turned on by default.
6. (Optional) In the **Summary** tab, review your selections.
7. Click **Generate** to generate the IP core and other selected deliverables.
8. In the **Review configuration generation** dialog box, click **Generate**. The Console in the **Summary** tab shows the generation status.

   Note: You can disable the **Review configuration generation** dialog box by turning off the **Show Confirmation Box** option in the wizard.
9. When generation finishes, the wizard displays the **Generation Success** dialog box. Click **OK** to close the wizard.

The wizard adds the IP to your project and displays it under **IP** in the Project pane.

Generated Files

The IP Manager generates these files and directories:

- **<module name>_define.vh**—Contains the customized parameters.
- **<module name>_tmpl.v**—Verilog HDL instantiation template.
- **<module name>_tmpl.vhd**—VHDL instantiation template.
- **<module name>.v**—IP source code.
- **settings.json**—Configuration file.
- **<kit name>_devkit**—Has generated RTL, example design, and Efinity® project targeting a specific development board.
- **Testbench**—Contains generated RTL and testbench files.

Note: Refer to the IP Manager chapter of the Efinity® Software User Guide for more information about the Efinity® IP Manager.

Customizing the AXI4-Stream Switch

The core has parameters so you can customize its function. You set the parameters in the General tab of the core's IP Configuration window.

Table 4: AXI4-Stream Switch Core Parameters

Parameter	Option	Description
Number of Slave Interfaces	1 – 8	Defines the number of slave interfaces connected to the master port. Default: 2
Number of Master Interfaces	1 – 8	Defines the number of master interfaces connected to the slave port. Default: 2
TDATA Width (bytes)	1 – 512	Defines the number of data bytes. Default: 1
Enable TLAST	Yes, No	Enable the TLAST signal in the AXI4-Stream port. Default: Yes
Enable TSTRB	Yes, No	Enable the TSTRB signal in the AXI4-Stream port. Default: No
Enable TKEEP	Yes, No	Enable the TKEEP signal in the AXI4-Stream port. Default: No
TDEST Widths (bits)	1 – 10	Defines the destination width. Default : 2
TID Width (bits)	0 – 32	Defines the ID width. Default : 0
TUSER Width (bits)	0 – 4096	Defines user width. Default : 0
Arbitration Mode	PRIORITY, ROUND_ROBIN_1	Defines the Arbitration Mode Default : ROUND_ROBIN_1
Arbitrate on Maximum Number of Transfer	4 – 1024	Defines the maximum of data byte transfer per transaction between arbitration. Default : 777
Arbitrate on Number of LOW TVALID Cycles	4 – 1024	Defines the continuous low TVALID per transaction between arbitration. Default : 777

Table 5: AXI4-Stream Switch Core Parameters (AXI Tab)

The number of AXI_S rows depends on the Number of Slave Interfaces parameter you set.

Parameters	Option	Description
MIN	0x00000000 – 0xFFFFF000	Defines the start of the slave destination address for each interface. Default: 0x00000000 (AXI_S0), 0x00000002 (AXI_S1)
MAX	0x00000000 – 0xFFFFF000	Defines the end of the slave destination address for each interface. Default: 0x00000001 (AXI_S0), 0x00000003 (AXI_S1)

AXI4-Stream Switch Example Design

You can choose to generate the example design when generating the core in the IP Manager Configuration window. Compile the example design project and download the **.hex** or **.bit** file to your board.

Important: 易灵思 tested the example design generated with the default parameter options only.

The example designs target the 钛金系列 Ti60 F225 Development Board. The design demonstrates four master ports accessing two different slave ports through the AXI4-Stream Switch core. The AXI4-Stream Switch core routes the read and write data based on the TDEST information.

Figure 2: AXI4-Stream Switch Example Design

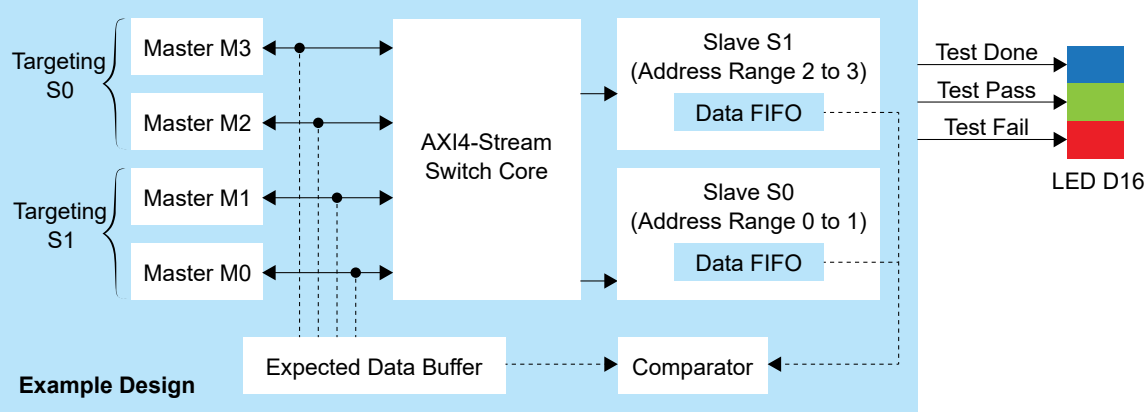


Table 6: Slave Ports Address Range

Slave	MIN	MAX
AXI_S0	0x00000000	0x00000001
AXI_S1	0x00000002	0x00000003

All four AXI master ports carry TDEST information between 0 to 3. Two of the four AXI masters (TDEST values 0 and 1) are routed to destination AXI_S0. While the other two (TDEST values 2 and 3) are routed to destination AXI_S1. Since there are two groups of master ports targeting different destination slave ports, the data transfer to the slave ports can happen simultaneously.

The **Enable TLAST** parameter is enabled in the example design. After the write operations are complete, the design compares the written data in the slave FIFOs to the expected data buffer and outputs the following on the development board LEDs:

Table 7: Example Design Output

Output		Description
LED D16 Blue	Test Done	Indicates the test is complete.
LED D16 Green	Test Pass	Indicates the written and read data are matched.
LED D16 Red	Test Fail	Indicates the written and read data are not matched.

AXI4-Stream Switch Testbench

You can choose to generate the testbench when generating the core in the IP Manager Configuration window.

Note: You must include all **.v** files generated in the **/testbench** directory in your simulation.

易灵思 provides a simulation script for you to run the testbench quickly using the Modelsim software. To run the Modelsim testbench script, run `vsim -do modelsim.do` in a terminal application. You must have Modelsim installed in your computer to use this script.

The testbench simulates the example design.

Revision History

Table 8: Revision History

Date	Version	Description
February 2023	1.1	Added note about the resource and performance values in the resource and utilization table are for guidance only.
August 2022	1.0	Initial release.