AXI FIFO
(Beta Release)

Version 1.0

May 15, 2023

Copyright

Trademarks

Disclaimers

Contents

IP Summary

Introduction

The AXI FIFO is an AXI full compliant customizable synchronous FIFO. It can be used to store and retrieve ordered data, while using optimal resources. AXI FIFO is derived from a native FIFO and made AXI4 compliant FIFO for easy integration with other AXI based systems.

Features

- 32-bit AXI4 slave interface
- Data width can be configured to 32, 64, 128, 256 or 512 bits for AXI4
- Configurable FIFO depth upto 512k

Overview

AXI FIFO

The AXI FIFO provides a buffer for storing write data until it can be read by the master, and provides read data to the master as soon as it becomes available. This allows the master to operate independently of the rate at which the slave can accept data.
The AXI FIFO works by using read and write pointers to store and access data in memory. When data is written to the FIFO, it is stored in a specific memory location based on the current write pointer. The write pointer then increments to indicate the next available memory location for future writes. Similarly, when data is read from the FIFO, the read pointer is used to access the memory location containing the next available data. A block diagram for the AXI-FIFO is shown in Figure 2.

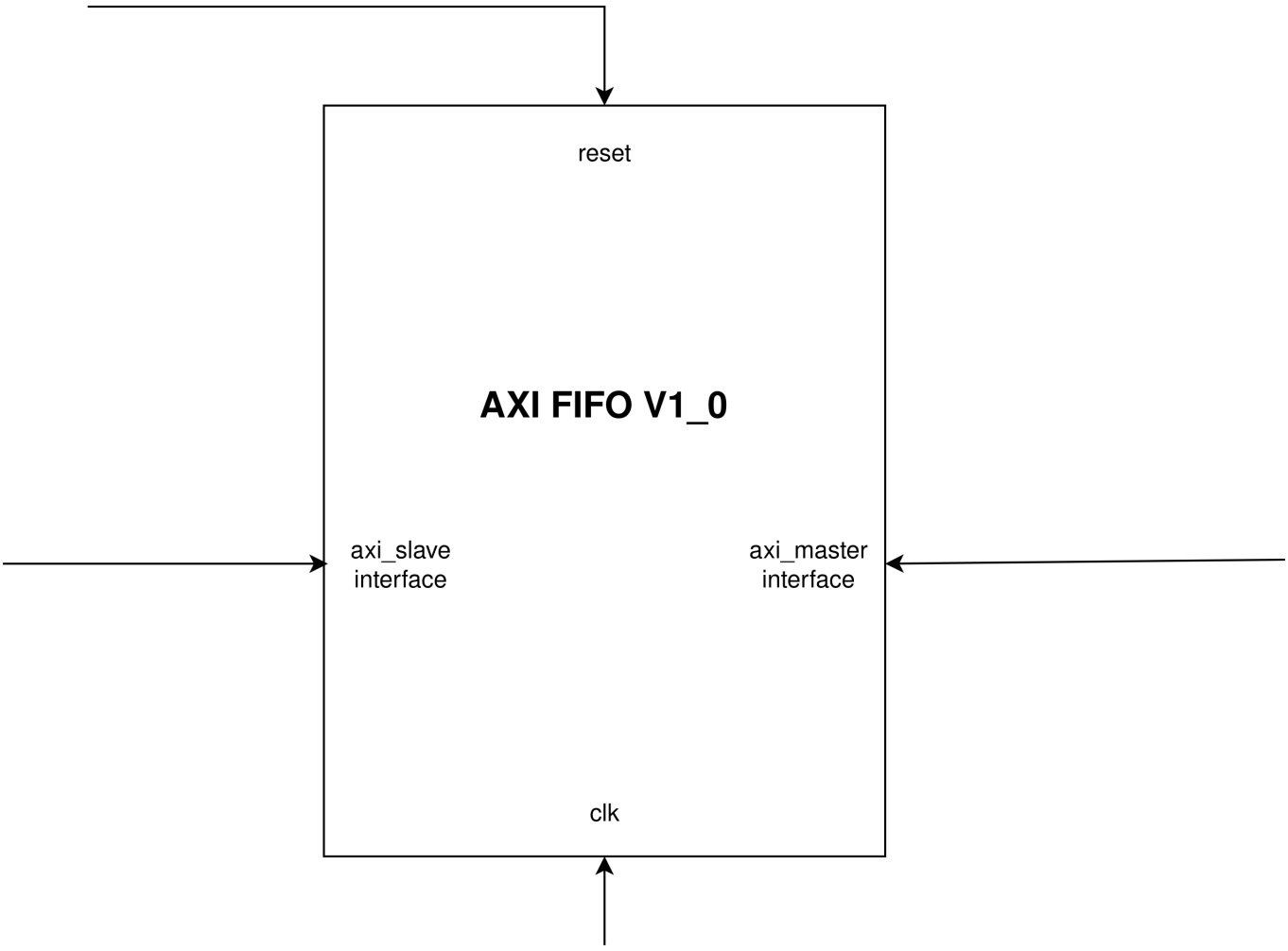

Figure 1: AXI FIFO Block Diagram

IP Specification

The AXI FIFO module has two main parts: the write path and the read path. The write path takes data from the AXI4 slave interface and stores it in a write data FIFO. The read path takes data from a read data FIFO and provides it to the AXI4 master interface.

In addition, the AXI FIFO is equipped with the flexibility to support various burst types and parametrizable data and address interface widths. Moreover, it offers the option to delay the address channel until either the write data is entirely shifted into the FIFO or the read data FIFO can accommodate the entire burst.

For the read channel, the AXI FIFO supports all burst types and provides the option to delay the address channel until either the read data FIFO is empty or has sufficient capacity to fit the whole burst. Similarly, for the write channel, the AXI FIFO supports all burst types and provides the option to delay the address channel until the write data is entirely shifted into the write data FIFO, or the current burst fills the write data FIFO completely. A block diagram for the AXI-FIFO is shown in Figure 2.

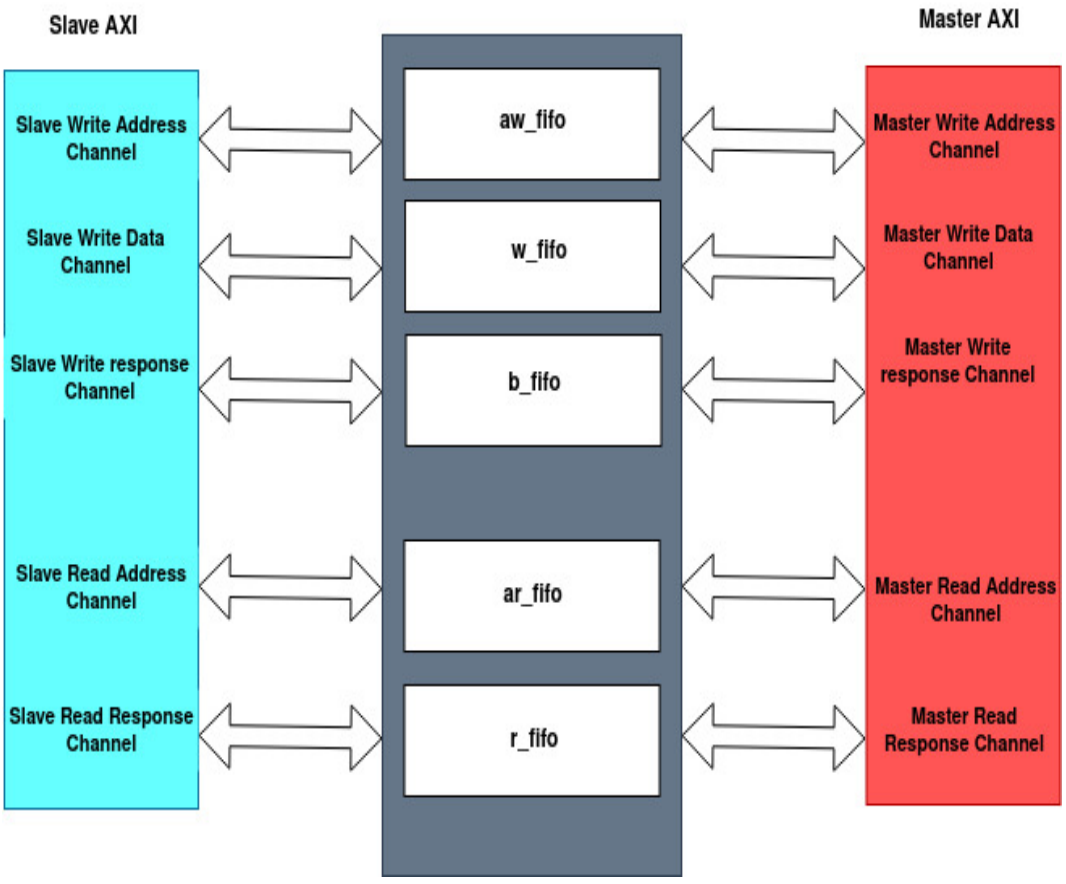

Figure 2: AXI FIFO Internal FIFO Diagram

Standards

The AXI4-Full interface is compliant with the AMBA® AXI Protocol Specification.

IP Support Details

The Table 1 gives the support details for AXI FIFO.

Compliance		IP Resources					Tool Flow	
Device	Interface	Source Files	Constraint File	Testbench	Simulation Model	Analyze and Elaboration	Simulation	Synthesis
GEMINI	AXI4 Full	Verilog	SDC	Python	CocoTb	Raptor(Verific)	Raptor Icarus	Raptor

Table 1: IP Details

Parameters

Table lists the parameters of the AXI FIFO.

Parameter	Values	Default Value	Description
DATA WIDTH	8,16,32,...,1024	32	Data width of data being transferred.
ADDR WIDTH	1-64	32	FIFO address width.
ID WIDTH	1-32	8	FIFO ID width.
AWUSER ENABLE	0-1	0	Depth of internal FIFO.
AWUSER WIDTH	1-1024	32	Data width of data being transferred.
WUSER ENABLE	0-1	0	FIFO address width.
WUSER WIDTH	1-1024	8	FIFO ID width.
BUSER ENABLE	0-1	0	Depth of internal FIFO.
BUSER WIDTH	1-1024	32	Data width of data being transferred.
ARUSER ENABLE	0-1	0	FIFO address width.
ARUSER WIDTH	1-1024	8	FIFO ID width.
RUSER ENABLE	0-1	0	Depth of internal FIFO.
RUSER WIDTH	1-1024	32	Data width of data being transferred.
WRITE FIFO DEPTH	0, 32, 512	32	FIFO address width.
READ FIFO DEPTH	0, 32, 512	32	FIFO ID width.
WRITE FIFO DELAY	0-1	0	Depth of internal FIFO.
READ FIFO DELAY	0-1	0	Data width of data being transferred.

AXI FIFO Parameters

Port List

Table 2 lists the top interface ports of the AXI FIFO.

Signal Name	I/O	Description
AXI Clock and Reset		
clk	I	AXI4-Lite Clock
rst	I	AXI4-Lite RESET
SLAVE INTERFACE		
AXI WRITE ADDRESS CHANNEL		
s_axil_awvalid	I	AXI4-Lite Write address valid
s_axil_awready	O	AXI4-Lite Write address ready
s_axil_awaddr	I	AXI4-Lite Write address
s_axil_awprot	I	AXI4-Lite Protection type
AXI WRITE DATA CHANNEL		
s_axil_wvalid	I	AXI4-Lite Write valid
s_axil_wready	O	AXI4-Lite Write ready.
s_axil_wdata	I	AXI4-Lite Write data
s_axil_wstrb	I	AXI4-Lite Write strobes
AXI WRITE RESPONSE CHANNEL		
s_axil_bvalid	O	AXI4-Lite Write response valid
s_axil_bready	I	AXI4-Lite Response ready
s_axil_bresp	O	AXI4-Lite Write response
AXI READ ADDRESS CHANNEL		
s_axil_arvalid	I	AXI4-Lite Read address valid
s_axil_arready	O	AXI4-Lite Read address ready
s_axil_araddr	I	AXI4-Lite Read address
s_axil_arprot	I	AXI4-Lite Protection type
AXI READ DATA CHANNEL		
s_axil_rvalid	I	AXI4-Lite Read valid
s_axil_rready	O	AXI4-Lite Read ready
s_axil_rresp	I	AXI4-Lite Read data
s_axil_rdata	O	AXI4-Lite Read response
MASTER INTERFACE		
AXI WRITE ADDRESS CHANNEL		
m_axil_awvalid	I	AXI4-Lite Write address valid
m_axil_awready	O	AXI4-Lite Write address ready
m_axil_awaddr	I	AXI4-Lite Write address
m_axil_awprot	I	AXI4-Lite Protection type
AXI WRITE DATA CHANNEL		
m_axil_wvalid	I	AXI4-Lite Write valid
m_axil_wready	O	AXI4-Lite Write ready.
m_axil_wdata	I	AXI4-Lite Write data
m_axil_wstrb	I	AXI4-Lite Write strobes

AXI WRITE RESPONSE CHANNEL		
m_axil_bvalid	O	AXI4-Lite Write response valid
m_axil_bready	I	AXI4-Lite Response ready
m_axil_bresp	O	AXI4-Lite Write response
AXI READ ADDRESS CHANNEL		
m_axil_arvalid	I	AXI4-Lite Read address valid
m_axil_arready	O	AXI4-Lite Read address ready
m_axil_araddr	I	AXI4-Lite Read address
m_axil_arprot	I	AXI4-Lite Protection type
AXI READ DATA CHANNEL		
m_axil_rvalid	I	AXI4-Lite Read valid
m_axil_rready	O	AXI4-Lite Read ready
m_axil_rresp	I	AXI4-Lite Read data
m_axil_rdata	O	AXI4-Lite Read response

Table 2: AXI FIFO Interface

Resource Utilization

The parameters for computing the maximum and minimum resource utilization are given in Table 3, remaining parameters have been kept at their default values.

Tool	Raptor Design Suite			
FPGA Device	GEMINI			
Configuration			**Resource Utilization**	
Minimum Resource	**Options**	**Configuration**	**Resource**	**Utilized**
	DATA WIDTH	32	BRAMs	3
	ADDR WIDTH	16	LUTs	209
	WRITE FIFO DEPTH	32	Registers	293
Maximum Resource	**Options**	**Configuration**	**Resource**	**Utilized**
	DATA WIDTH	1024	BRAMs	62
	ADDR WIDTH	32	LUTs	255
	WRITE FIFO DEPTH	512	Registers	2426

Table 3: Resource Utilization

Design Flow

IP Customization and Generation

AXI FIFO IP core is a part of the Raptor Design Suite Software. A customized AXI FIFO can be generated from the Raptor's IP configurator window as shown in Figure 3.

Figure 3: IP list

Parameters Customization

From the IP configuration window, the parameters of the AXI FIFO can be configured and AXI FIFO features can be enabled for generating a customized AXI FIFO IP core that suits the user application requirement as shown in Figure 4. After IP Customization, all the source files are made available to the user with a top wrapper that instantiates a parameterized instance of the AXI FIFO.

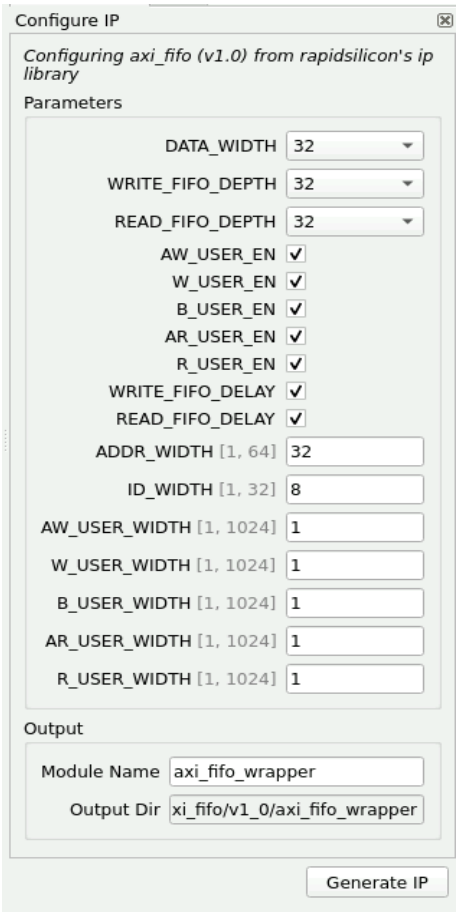

Figure 4: IP Configuration

Synthesis and PR

Raptor Suite is armed with tools for Synthesis along with Post and Route capabilities and the generated post-synthesis and post-route and place net-lists can be viewed and analyzed from within the Raptor. The generated bit-stream can then be uploaded on an FPGA device to be utilized in hardware applications.

Test Bench

The AXI FIFO simulation is based on Cocotb. It has a complete environment that extensively tests AXI FIFO as a DUT. It has 25 tests in total, 12 write tests, 12 read tests and a stress test.

The simulation can be run from Raptor IP Catalog. User can interact with the wavefrom from within Raptor.

Release

Release History

Date	Version	Revisions
May 15, 2023	1.0	Initial version AXI4-FIFO User Guide Document