



# **AXI4 Interconnect (Beta Release)**

Version 0.1

May 10, 2023

## Copyright

Copyright © 2021 Rapid Silicon. All rights reserved. This document may not, in whole or part, be reproduced, modified, distributed, or publicly displayed without prior written consent from Rapid Silicon ("Rapid Silicon").

## Trademarks

All Rapid Silicon trademarks are as listed at [www.rapidsilicon.com](http://www.rapidsilicon.com). Synopsys and Synplify Pro are trademarks of Synopsys, Inc. Aldec and Active-HDL are trademarks of Aldec, Inc. Modelsim and Questa are trademarks or registered trademarks of Siemens Industry Software Inc. or its subsidiaries in the United States or other countries. All other trademarks are the property of their respective owners.

## Disclaimers

NO WARRANTIES: THE INFORMATION PROVIDED IN THIS DOCUMENT IS "AS IS" WITHOUT ANY EXPRESS OR IMPLIED WARRANTY OF ANY KIND INCLUDING WARRANTIES OF ACCURACY, COMPLETENESS, MERCHANTABILITY, NONINFRINGEMENT OF INTELLECTUAL PROPERTY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL RAPID SILICON OR ITS SUPPLIERS BE LIABLE FOR ANY DAMAGES WHATSOEVER (WHETHER DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL, INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OF OR INABILITY TO USE THE INFORMATION PROVIDED IN THIS DOCUMENT, EVEN IF RAPID SILICON HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME JURISDICTIONS PROHIBIT THE EXCLUSION OR LIMITATION OF CERTAIN LIABILITY, SOME OF THE ABOVE LIMITATIONS MAY NOT APPLY TO YOU.

Rapid Silicon may make changes to these materials, specifications, or information, or to the products described herein, at any time without notice. Rapid Silicon makes no commitment to update this documentation. Rapid Silicon reserves the right to discontinue any product or service without notice and assumes no obligation to correct any errors contained herein or to advise any user of this document of any correction if such be made. Rapid Silicon recommends its customers obtain the latest version of the relevant information to establish that the information being relied upon is current and before ordering any products.

# Contents

<b>IP Summary</b>	<b>3</b>
Introduction . . . . .	3
Features . . . . .	3
<b>Overview</b>	<b>4</b>
AXI4 Interconnect . . . . .	4
<b>IP Specification</b>	<b>5</b>
Address Decoding . . . . .	6
Arbitration . . . . .	6
Standards . . . . .	7
IP Support Details . . . . .	7
Resource Utilization . . . . .	7
Port List . . . . .	8
Parameters . . . . .	10
<b>Design Flow</b>	<b>11</b>
IP Customization and Generation . . . . .	11
Parameters Customization . . . . .	11
<b>Test Bench</b>	<b>12</b>
<b>Revision History</b>	<b>13</b>

# IP Summary

## Introduction

The AXI4 (Advanced eXtensible Interface 4) interconnect is a widely-used, industry-standard protocol for connecting intellectual property (IP) blocks in a system-on-chip (SoC) design. The AXI4 interconnect supports high-bandwidth, low-latency communication between IP blocks, and provides a range of features and functionality to optimize system performance and reduce design complexity. It includes separate read and write channels for data and control information, support for burst transfers and out-of-order transaction processing, and features to support cache coherency and multi-master configurations. The AXI4 interconnect is widely used in a range of applications, including mobile devices, networking equipment, and high-performance computing systems. It provides a standardized interface that allows IP blocks from different vendors to be easily integrated into a single SoC design, reducing development time and cost.

## Features

- High performance: AXI4 interconnect is designed for high performance with a high-bandwidth, low-latency interface that can handle large amounts of data.
- Scalability: The AXI4 interconnect is highly scalable, supporting a large number of masters and slaves. This makes it suitable for complex SoC designs.
- Burst transfers: The AXI4 interconnect supports burst transfers, which allows for more efficient data transfer by reducing the number of transactions required.
- Configurability: AXI4 interconnect is highly configurable, allowing designers to optimize the interconnect for their specific SoC design.
- Address and data interleaving: AXI4 interconnect supports address and data interleaving, which allows for faster data transfers by overlapping address and data phases.
- Master Count: 16
- Slave Count: 16
- Data Width: 8, 16, 32, 64, 128, 256 bits
- Address Width: 32, 64, 128 bits
- User Width (per channel): Up to 1024 bits
- ID Width: Up to 8 bits

# Overview

## AXI4 Interconnect

AXI Interconnect IP core is a component used in system-on-chip (SoC) designs to connect multiple AXI masters and slaves. It acts as a central hub or router that interconnects the AXI components in a system and provides a common communication protocol for them. This IP core supports the AXI protocol. It includes multiple AXI slave and master ports, enabling it to connect multiple AXI components within a system. It uses arbitration and routing logic to manage the data transfers between the AXI components connected to its ports. It also supports various routing schemes, such as round-robin and fixed priority. The block diagram of AXI4 Interconnect is given in figure 1.

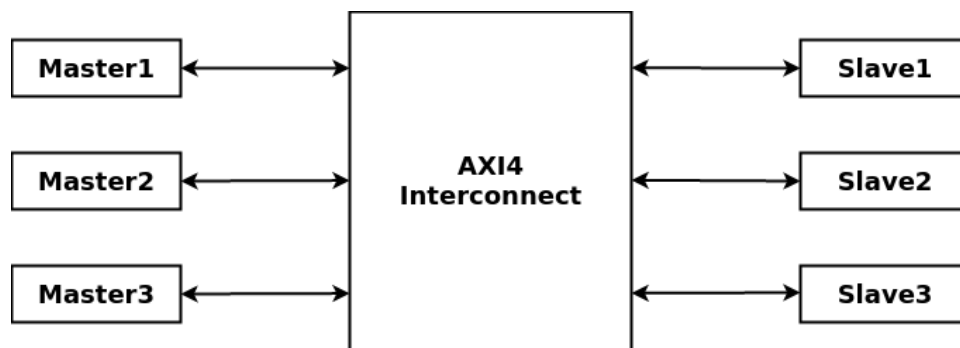


Figure 1: AXI4 Interconnect Block Diagram

# IP Specification

The AXI4 Interconnect IP specification provides a standardized approach for connecting different components of a system-on-chip, such as processors, memories, DMA controllers, and other IP blocks, through a common bus architecture. It provides a set of rules and protocols for data transfer, flow control, arbitration, and other aspects of communication between the components. It supports multiple masters and multiple slaves, and can be configured to support different data widths, burst sizes, and transfer modes. It also supports various QoS (Quality of Service) levels and power management features to optimize system performance and energy efficiency. The figure 2 shows the top level diagram of AXI4 Interconnect.

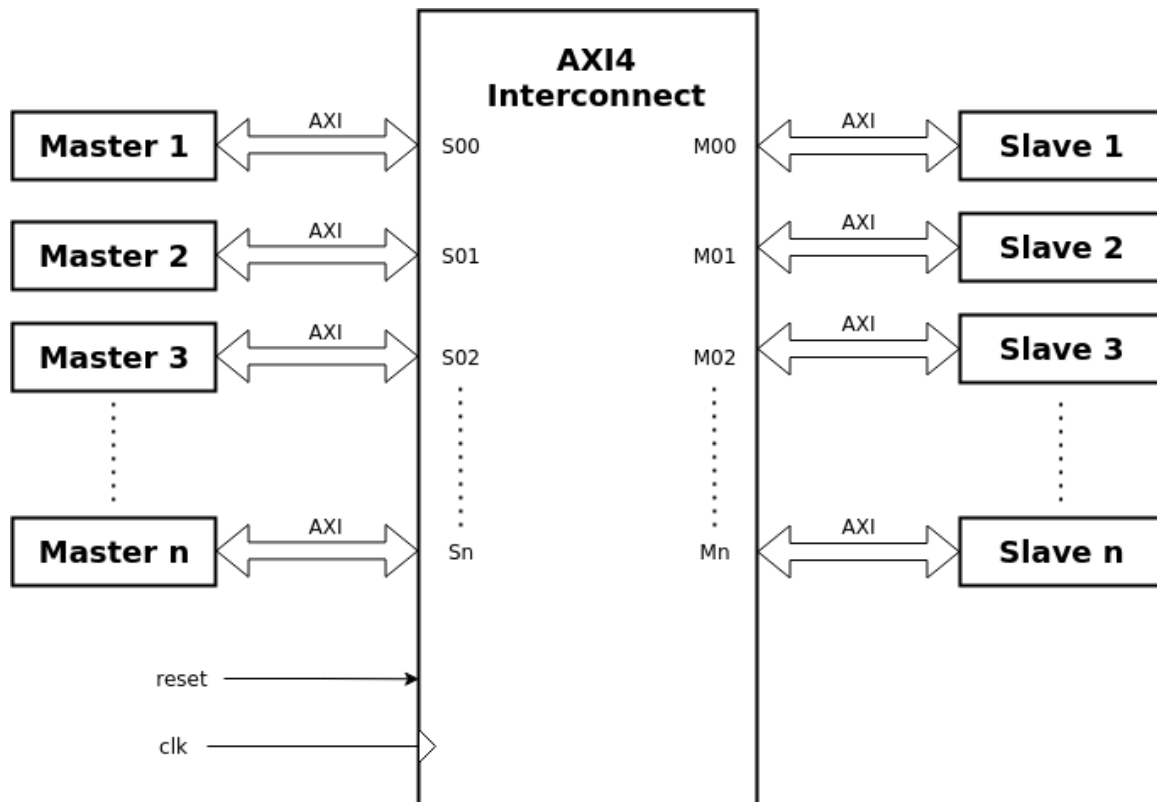


Figure 2: Top Module

## Address Decoding

The block diagram in Figure 3 depicts the Address Decoder responsible for assigning address space to each Master Interface and maintaining an address table. When a transaction occurs on the Address Write or Address Read channels of the Slave Interface, the AXI4 Interconnect core must determine which Master Interface is the intended target by decoding the address. The address decoder ensures that there is no overlapping of addresses and that all addresses are aligned. It follows certain rules while assigning address space to each master interface. If the address received on the Slave Interface does not match any of the address ranges being decoded by the Address Decoder, the transaction is trapped and handled by a decoded error module. In such situations, the interconnect generates a protocol compliant response indicating that there is no available slave within this address range.

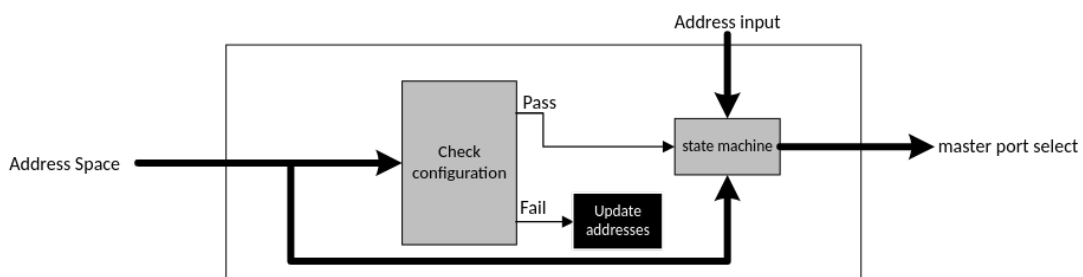


Figure 3: Address Decoding

## Arbitration

The AXI4 Interconnect core has arbiters for both write and read channel instances, which are responsible for address and response arbitration. The priority encoder sets the relative priority for each transaction, and arbitration among different transactions is decided using round-robin methodology. In a write transaction, both AWVALID and AWREADY signals must be high for the transaction to start, and the transaction is considered complete when a BVALID/BREADY handshake is completed. The counter increments upon the start of a transaction and decrements upon its completion, enabling the calculation of the total number of write transactions in flight. In a read transaction, both ARVALID and ARREADY signals go high when the transaction starts, while transactions are considered complete when an RVALID/RREADY handshake completes with RLAST asserted. The Interconnect counts the total number of read transactions in flight based on these signals. Transactions that target a master interface that has reached its issuing limit are disqualified from arbitration, and their request is not forwarded to the arbiter.

## Standards

The AXI4 Master and Slave interfaces are compliant with the AMBA® AXI Protocol Specification.

## IP Support Details

The Table 1 gives the support details for AXI4 Interconnect.

Compliance		IP Resources				Tool Flow		
Device	Interface	Source Files	Constraint File	Testbench	Simulation Model	Analyze and Elaboration	Simulation	Synthesis
GEMINI	AXI4	Verilog	-	Cocotb	-	Raptor	Raptor	Raptor

Table 1: Support Details

## Resource Utilization

The parameters for computing the maximum and the minimum resource utilization are given in Table 2, remaining parameters have been kept at their default values.

Tool	Raptor Design Suite			
FPGA Device	GEMINI			
Minimum Resource	Configuration		Resource Utilization	
	Options	Configuration	Resources	Utilized
	S_COUNT	1	BRAMS	0
	M_COUNT	1	REGISTERS	152
	DATA_WIDTH	8	LUTS	173
	ADDR_WIDTH	32	-	-
Maximum Resources	Options	Configuration	Resources	Utilized
	S_COUNT	16	BRAMS	3
	M_COUNT	16	REGISTERS	1521
	DATA_WIDTH	256	LUTS	6212
	ADDR_WIDTH	128	-	-

Table 2: Resource Utilization



## Ports

Table 3 lists the top interface ports of the AXI4 Interconnect and 'x' represents the number of interface.

Signal Name	I/O	Description
clk	I	Clock Signal for Interconnect
rst	I	Active High Synchronous Reset Signal
<b>Master Write Address Channel</b>		
s<x>_axi_awid	I	Write address ID
s<x>_axi_awaddr	I	Write address
s<x>_axi_awlen	I	Burst length
s<x>_axi_awsz	I	Burst size
s<x>_axi_awburst	I	Burst type
s<x>_axi_awlock	I	Lock type
s<x>_axi_awcache	I	Memory type
s<x>_axi_awprot	I	Protection type
s<x>_axi_awvalid	I	Write address valid
s<x>_axi_awready	O	Write address ready
<b>Master Write Data Channel</b>		
s<x>_axi_wdata	I	Write data
s<x>_axi_wstrb	I	Write strobe
s<x>_axi_wlast	I	Write last
s<x>_axi_wvalid	I	Write valid
s<x>_axi_wready	O	Write ready
<b>Master Write Response Channel</b>		
s<x>_axi_bid	O	Response ID tag
s<x>_axi_bresp	O	Write response
s<x>_axi_bvalid	O	Write response valid
s<x>_axi_bready	I	Write response ready
<b>Master Read Address Channel</b>		
s<x>_axi_arid	I	Read address ID
s<x>_axi_araddr	I	Read address
s<x>_axi_arlen	I	Burst length
s<x>_axi_arsz	I	Burst size
s<x>_axi_arburst	I	Burst type
s<x>_axi_arlock	I	Lock type
s<x>_axi_arcache	I	Memory type
s<x>_axi_arprot	I	Protection type
s<x>_axi_arvalid	I	Read address valid
s<x>_axi_arready	O	Read address ready
<b>Master Read Data Channel</b>		
s<x>_axi_rid	O	Read ID tag
s<x>_axi_rdata	O	Read data

Signal Name	I/O	Description
s<x>_axi_rresp	0	Read response
s<x>_axi_rlast	0	Read last
s<x>_axi_rvalid	0	Read valid
s<x>_axi_rready	1	Read ready
<b>Slave Write Address Channel</b>		
m<x>_axi_awid	0	Write address ID
m<x>_axi_awaddr	0	Write address
m<x>_axi_awlen	0	Burst length
m<x>_axi_awsz	0	Burst size
m<x>_axi_awburst	0	Burst type
m<x>_axi_awlock	0	Lock type
m<x>_axi_awcache	0	Memory type
m<x>_axi_awprot	0	Protection type
m<x>_axi_awvalid	0	Write address valid
m<x>_axi_awready	1	Write address ready
<b>Slave Write Data Channel</b>		
m<x>_axi_wdata	0	Write data
m<x>_axi_wstrb	0	Write strobe
m<x>_axi_wlast	0	Write last
m<x>_axi_wvalid	0	Write valid
m<x>_axi_wready	1	Write ready
<b>Slave Write Response Channel</b>		
m<x>_axi_bid	1	Response ID tag
m<x>_axi_bresp	1	Write response
m<x>_axi_bvalid	1	Write response valid
m<x>_axi_bready	0	Write response ready
<b>Slave Read Address Channel</b>		
m<x>_axi_arid	0	Read address ID
m<x>_axi_araddr	0	Read address
m<x>_axi_arlen	0	Burst length
m<x>_axi_arsz	0	Burst size
m<x>_axi_arburst	0	Burst type
m<x>_axi_arlock	0	Lock type
m<x>_axi_arcache	0	Memory type
m<x>_axi_arprot	0	Protection type
m<x>_axi_arvalid	0	Read address valid
m<x>_axi_arready	1	Read address ready
<b>Slave Read Data Channel</b>		
m<x>_axi_rid	1	Read ID tag
m<x>_axi_rdata	1	Read data
m<x>_axi_rresp	1	Read response
m<x>_axi_rlast	1	Read last

Signal Name	I/O	Description
m<x>_axi_rvalid	I	Read valid
m<x>_axi_rready	O	Read ready

Table 3: Port List

## Parameters

Table 4 lists the parameters of the AXI4 Interconnect.

Parameter	Values	Default Value	Description
S_COUNT	1-16	4	No. of Slaves connected to Interconnect
M_COUNT	1-16	4	No. of Masters connected to Interconnect
DATA_WIDTH	8, 16, 32, 64, 128, 256	32	Data Width of Interconnect
ADDR_WIDTH	32, 64, 128	32	Address Width of Interconnect
ID_WIDTH	1-1024	1	ID field of Interconnect
AW_USER_EN	True/False	True	User Enable Field for AW Channel
W_USER_EN	True/False	True	User Enable Field for W Channel
B_USER_EN	True/False	True	User Enable Field for B Channel
AR_USER_EN	True/False	True	User Enable Field for AR Channel
R_USER_EN	True/False	True	User Enable Field for R Channel
AW_USER_WIDTH	1-1024	1	User Field for AW Channel
W_USER_WIDTH	1-1024	1	User Field for W Channel
B_USER_WIDTH	1-1024	1	User Field for B Channel
AR_USER_WIDTH	1-1024	1	User Field for AR Channel
R_USER_WIDTH	1-1024	1	User Field for R Channel

Table 4: Parameters

# Design Flow

## IP Customization and Generation

AXI4 Interconnect IP core is a part of the Raptor Design Suite Software. A customized interconnect can be generated from the Raptor's IP configuration window as shown in figure 4.

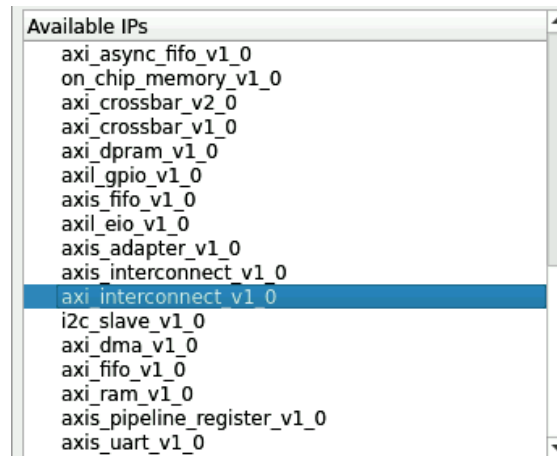


Figure 4: IP List

## Parameters Customization

From the IP configuration window, the parameters of the AXI4 Interconnect can be configured and its features can be enabled for generating a customized IP core that suits the user application requirements. All parameters are shown in figure 5.

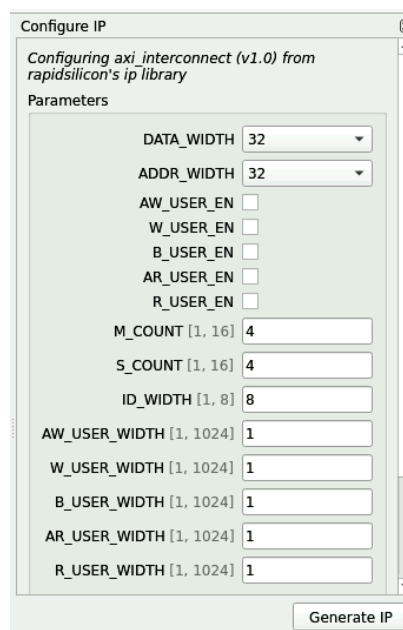


Figure 5: IP Configuration

# Test Bench

The AXI4 interconnect IP is provided with a testbench which is based upon Coco TB verification environment. In this test, four masters and four slaves are connected to interconnect. Interconnect assigns fixed address space to each slave. Each master communicates with each slave. The input data is generated using a test data generator module. Input data is routed from master to slave through interconnect. The output data is compared with the expected output data to verify the correctness of the IP core's operation. The dump file is generated to view the output of the test. In the end, there is status for passing or failure of the test.

# Revision History

Date	Version	Revisions
May 10, 2023	0.1	Initial version AXI4 Interconnect User Guide