# AXI4 Interconnect

Version 1.0

May 4, 2023

## Copyright



## Trademarks



## Disclaimers

# Contents

# 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.

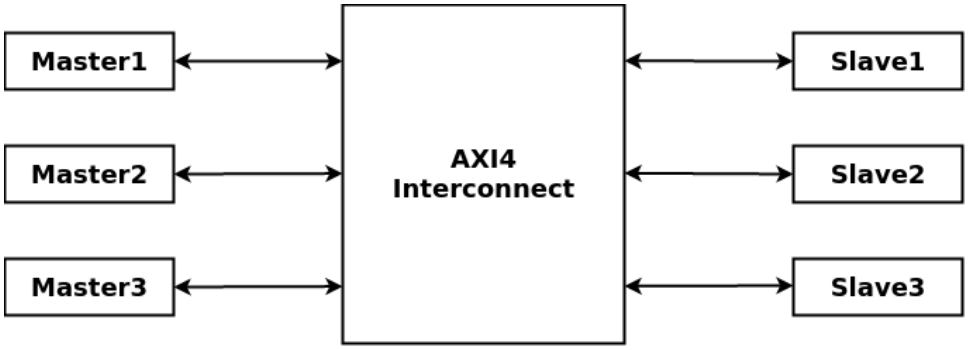


Figure 1: AXI4 Interconnect Block Diagram

# IP Specification

## Overview

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.

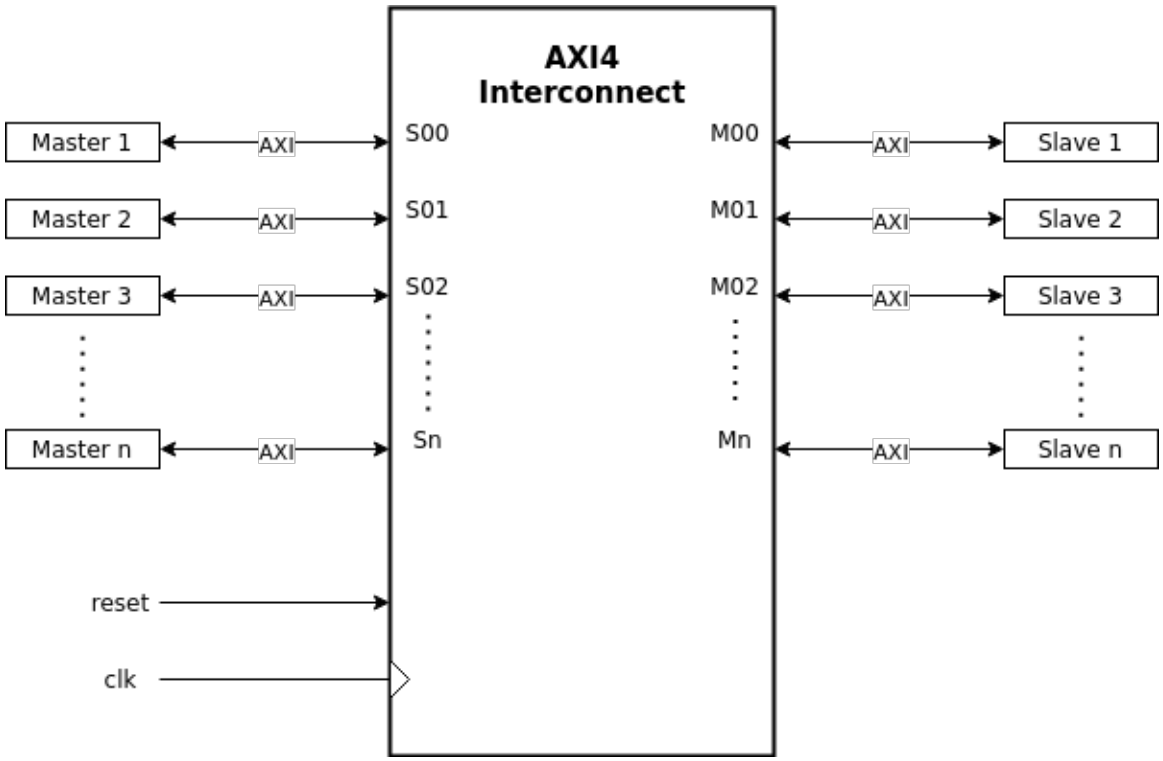


Figure 2: Top Module

## 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 | | | |
| **Configuration** | | | **Resource Utilization** | |
| Minimum Resource | **Options** | **Configuration** | **Resources** | **Utilized** |
| | S_COUNT | 1 | BRAMS | 0 |
| | M_COUNT | 1 | REGISTERS | 152 |
| | DATA_WIDTH | 8 | LUTS | 173 |
| | ADDR_WIDTH | 32 | - | - |
| Maximum Resource | **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.

| Signal Name | Input/Output | Description |
|---|---|---|
| clk | I | Clock Signal of Interconnect |
| rst | I | Active Low Reset Signal |
| **Master Write Address Channel** | | |
| s_axi_awid | I | Write address ID |
| s_axi_awaddr | I | Write address |
| s_axi_awlen | I | Burst length |
| s_axi_awsize | I | Burst size |
| s_axi_awburst | I | Burst type |
| s_axi_awlock | I | Lock type |
| s_axi_awcache | I | Memory type |
| s_axi_awprot | I | Protection type |
| s_axi_awvalid | I | Write address valid |
| s_axi_awready | O | Write address ready |
| **Master Write Data Channel** | | |
| s_axi_wdata | I | Write data |
| s_axi_wstrb | I | Write strobe |
| s_axi_wlast | I | Write last |
| s_axi_wvalid | I | Write valid |
| s_axi_wready | O | Write ready |
| **Master Write Response Channel** | | |
| s_axi_bid | O | Response ID tag |
| s_axi_bresp | O | Write response |
| s_axi_bvalid | O | Write response valid |
| s_axi_bready | I | Write response ready |
| **Master Read Address Channel** | | |
| s_axi_arid | I | Read address ID |
| s_axi_araddr | I | Read address |

| Signal Name | Input/Output | Description |
|---|---|---|
| s_axi_arlen | I | Burst length |
| s_axi_arsize | I | Burst size |
| s_axi_arburst | I | Burst type |
| s_axi_arlock | I | Lock type |
| s_axi_arcache | I | Memory type |
| s_axi_arprot | I | Protection type |
| s_axi_arvalid | I | Read address valid |
| s_axi_arready | O | Read address ready |
| **Master Read Data Channel** | | |
| s_axi_rid | O | Read ID tag |
| s_axi_rdata | O | Read data |
| s_axi_rresp | O | Read response |
| s_axi_rlast | O | Read last |
| s_axi_rvalid | O | Read valid |
| s_axi_rready | I | Read ready |
| **Slave Write Address Channel** | | |
| m_axi_awid | O | Write address ID |
| m_axi_awaddr | O | Write address |
| m_axi_awlen | O | Burst length |
| m_axi_awsize | O | Burst size |
| m_axi_awburst | O | Burst type |
| m_axi_awlock | O | Lock type |
| m_axi_awcache | O | Memory type |
| m_axi_awprot | O | Protection type |
| m_axi_awvalid | O | Write address valid |
| m_axi_awready | I | Write address ready |
| **Slave Write Data Channel** | | |
| m_axi_wdata | O | Write data |
| m_axi_wstrb | O | Write strobe |

| Signal Name | Input/Output | Description |
|---|---|---|
| m_axi_wlast | O | Write last |
| m_axi_wvalid | O | Write valid |
| m_axi_wready | I | Write ready |
| **Slave Write Response Channel** | | |
| m_axi_bid | I | Response ID tag |
| m_axi_bresp | I | Write response |
| m_axi_bvalid | I | Write response valid |
| m_axi_bready | O | Write response ready |
| **Slave Read Address Channel** | | |
| m_axi_arid | O | Read address ID |
| m_axi_araddr | O | Read address |
| m_axi_arlen | O | Burst length |
| m_axi_arsize | O | Burst size |
| m_axi_arburst | O | Burst type |
| m_axi_arlock | O | Lock type |
| m_axi_arcache | O | Memory type |
| m_axi_arprot | O | Protection type |
| m_axi_arvalid | O | Read address valid |
| m_axi_arready | I | Read address ready |
| **Slave Read Data Channel** | | |
| m_axi_rid | I | Read ID tag |
| m_axi_rdata | I | Read data |
| m_axi_rresp | I | Read response |
| m_axi_rlast | I | Read last |
| m_axi_rvalid | I | Read valid |
| m_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 3.

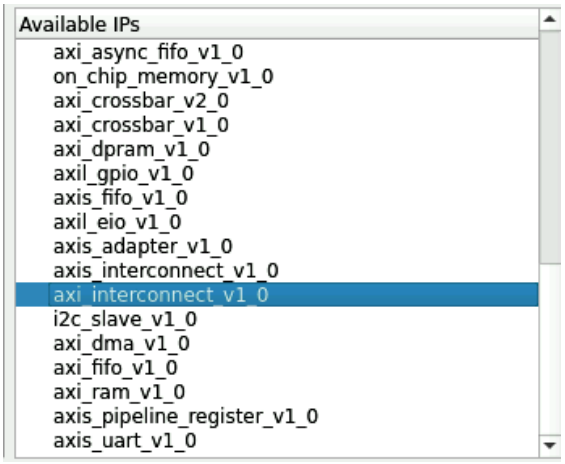


Figure 3: IP List

## Parameters Customization

From the IP configuration window, the parameters of the AXI4 Interconnect can be configured and it's features can be enabled for generating a customized IP core that suits the user application requirements. All parameters are shown in figure 4.

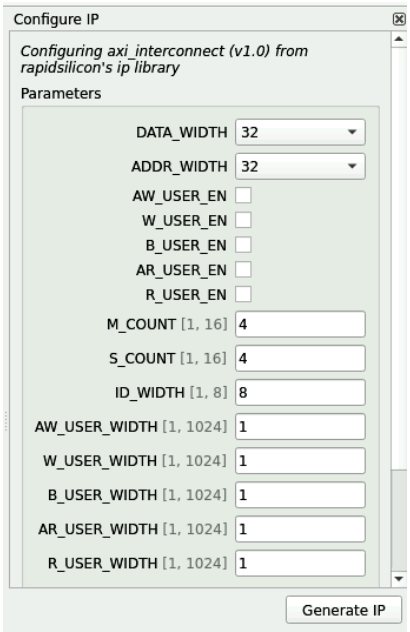


Figure 4: IP Configuration

# Test Bench

## Test for AXI4 Interconnect 4x4

The testbench attached with AXI4 Interconnect is CocoTB based verification environment. In this test, four masters ad four slaves are connected to interconnect. Interconnect assigns address space to each slave. Each master communicate 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 4, 2023 | 1.0 | Initial version AXI4 Interconnect User Guide |