

AXIL Ethernet v1.0

IP User Guide (Beta Release)



January 20, 2023

Contents

IP Summary	2
Overview	3
AXIL Ethernet	3
Licensing	4
IP Specification	5
Standards	5
IP Support Details	6
Resource Utilization	6
Parameters	6
Port List	7
Design Flow	8
IP Customization and Generation	8
Example Design	10
Overview	10
Simulating the Example Design	10
Synthesis and PnR	10
Test Bench	11
Release	12
Revision History	12

IP Summary

Introduction

AXI Lite Ethernet is a type of AXI (Advanced eXtensible Interface) peripheral that connects to the AXI bus of a system on a chip (SoC) and provides Ethernet connectivity. AXI Lite Ethernet is a simple version of the AXI Ethernet, it is designed for SoC and FPGA (Field-Programmable Gate Array) that have limited resources. This IP is especially useful when the configuration of Ethernet needs to be modified at run-time as it provided various different configurations useful for different types of scenarios of data transfer.

Features

- Supports Mii or Model Phy for both emulation and simulation.
- Configurable TX Slots.
- Configurable RX Slots.
- Configurable Bus Endianness.
- Included Management Data Interface for complete functionality.

Overview

AXIL Ethernet

AXIL Ethernet is a communication interface used to connect an Ethernet MAC (Media Access Control) to an AXI bus. AXI Lite Ethernet is a simpler version of AXI Ethernet and does not provide all the features of the full AXI Ethernet peripheral. It is typically used in embedded systems and FPGAs where resources are limited. Ethernet is a family of wired networking technologies and can support a vast list of bit rates over a multitude of frequencies. The Ethernet standards include several wiring and signaling variants of the OSI physical layer, or simply, PHY. This IP core provides both the Mii and Model configuration for the PHY, making it suitable for both real world uses and for simulatory purposes. A macro block diagram of this AXILite Ethernet Core connected inside a DUT is shown in Figure 1.

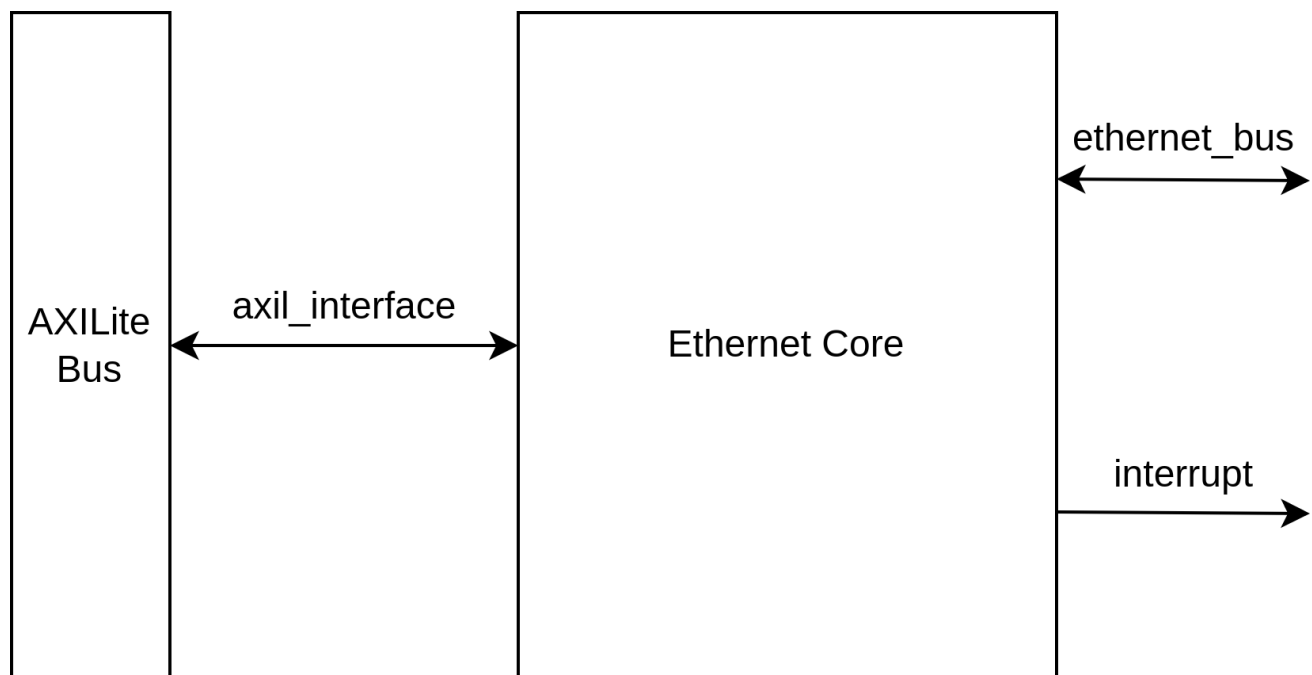


Figure 1. AXIL Ethernet Block Diagram

Licensing

COPYRIGHT TEXT:

Copyright (c) 2022 RapidSilicon

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

IP Specification

Ethernet is a type of computer networking technology that uses a physical cable to connect devices and transmit data. It uses a protocol called the Media Access Control (MAC) to govern how devices on the network access and transmit data over the cable. Ethernet uses the TCP/IP stack to converse between different devices and their networks. When a device wants to transmit data, it first listens to the network to ensure that no other device is currently transmitting. If the network is clear, the device sends its data, which is then received by all other devices on the network. If two devices attempt to transmit at the same time, a collision occurs, and both devices must wait and re-transmit their data at a later time. This Ethernet IP features configurable RX / TX slots as well as configurable bus endianness. It supports the **media-independent interface** (MII) standard. This MII can be used to connect the embedded MAC to an external PHY. This IP core uses the AXILite interface for the data transmission with a Master, such as in an SoC, while using the ethernet protocol for the TCP/IP data transmission and recieval. It consists of various error notifying signals too such as Receiver Error, Transmission Enable, Interrupt while also sporting bidirectional Management Data Interface and its clock.

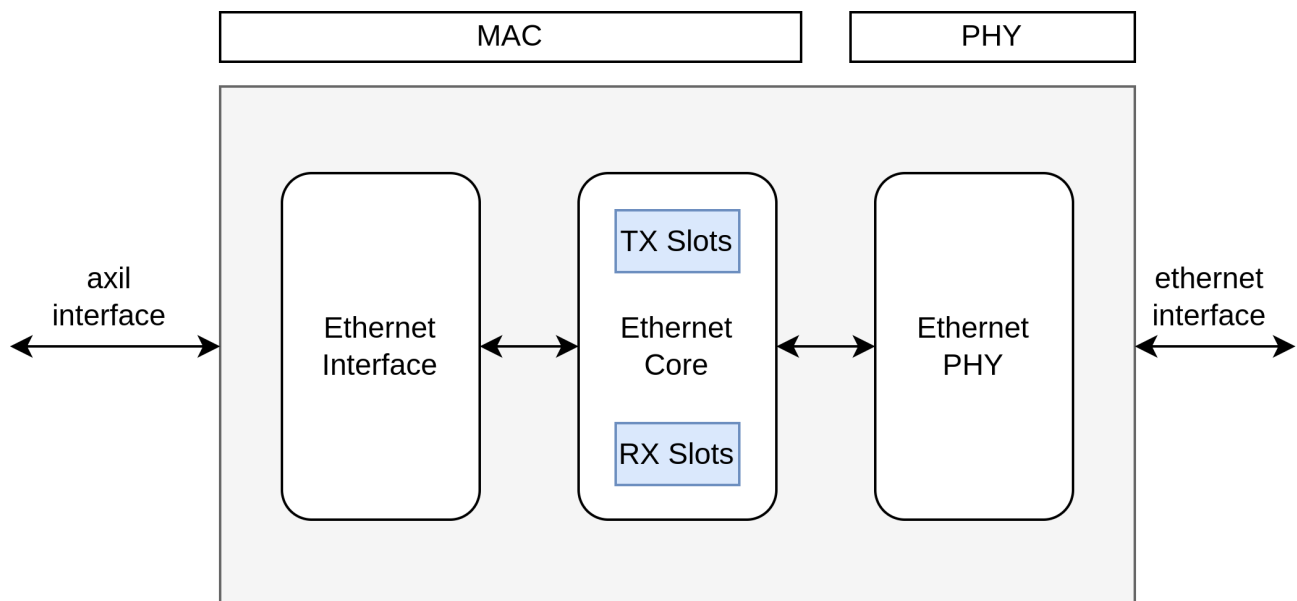


Figure 2. Top Module

Standards

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

IP Support Details

The Table 1 gives the support details for AXIL Ethernet.

Compliance		IP Resources					Tool Flow		
Device	Interface	Source Files	Constraint File	Testbench	Simulation Model	Software Driver	Analyze and Elaboration	Simulation	Synthesis
GEMINI	AXI4-Lite	Verilog	SDC	Python / C	LiteX	Verilator	Raptor	Raptor	Raptor

Table 1. IP Details

Resource Utilization

The parameters for computing the maximum and 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
	Core NTXSlots	1	LUT	1192
	Core NRXSlots	1	Registers	872
	Core Bus Endianness	Big	BRAM	11
Maximum Resource	Options	Configuration	Resources	Utilized
	Core NTXSlots	4	LUT	1195
	Core NRXSlots	4	Registers	872
	Core Bus Endianness	Little	BRAM	11

Table 2. AXIL Ethernet Resource Utilization

Parameters

Table 3 lists the parameters of the AXIL Ethernet.

Parameter	Values	Default Value	Description
Core PHY	Mii / Model	Mii	Type of PHY
Core NTXSlots	1, 2, 4	2	Number of TX Slots
Core RTXSlots	1, 2, 4	2	Number of RX Slots
Core Bus Endianness	Big / Little	Big	Bus Endianness

Table 3. Parameters

Ports

Table 4 lists the top interface ports of the AXIL Ethernet.

Signal Name	I/O	Description
AXI Clock and Reset		
sys_clk	I	AXI4-Lite Clock
sys_reset	I	AXI4-Lite RESET
AXI Write Address Channel		
bus_aw_valid	I	AXI4-Lite Write Address Valid
bus_aw_prot	I	AXI4-Lite Write Address Protection type
bus_aw_addr	I	AXI4-Lite Write Address
bus_aw_ready	O	AXI4-Lite Write Address Ready
AXI Write Channel		
bus_w_valid	I	AXI4-Lite Write Data Valid
bus_w_ready	O	AXI4-Lite Write Data Ready
bus_w_data	I	AXI4-Lite Write Data
bus_w_strb	I	AXI4-Lite Write Data Strobe
AXI Write Response Channel		
bus_b_valid	O	AXI4-Lite Write Response Valid
bus_b_ready	I	AXI4-Lite Write Response Ready
bus_b_resp	O	AXI4-Lite Write Response
AXI Read Address Channel		
bus_ar_valid	I	AXI4-Lite Read Address Valid
bus_ar_ready	O	AXI4-Lite Read Address Ready
bus_ar_addr	I	AXI4-Lite Read Address
bus_ar_prot	I	AXI4-Lite Read Address Protection Type
AXI Read Channel		
bus_r_valid	O	AXI4-Lite Read Data Valid
bus_r_ready	I	AXI4-Lite Read Data Ready
bus_r_resp	O	AXI4-Lite Read Data Response
bus_r_data	O	AXI4-Lite Read Data
Ethernet Signals		
mii_eth_clocks_tx	I	Ethernet Transmission Clock
mii_eth_clocks_rx	I	Ethernet Receiver Clock
mii_eth_rst_n	O	Ethernet Active High Reset
mii_eth_mdio	I / O	Ethernet Management Data Input / Output
mii_eth_mdc	O	Ethernet Management Data Clock
mii_eth_rx_dv	I	Ethernet Receive Data Valid
mii_eth_rx_er	I	Ethernet Receive Error
mii_eth_rx_data	I	Ethernet Receive Data
mii_eth_tx_en	O	Ethernet Transmit Enable
mii_eth_tx_data	O	Ethernet Transmit Data
mii_eth_col	I	Ethernet Collision Detect
mii_eth_crs	I	Ethernet Carrier Sense
interrupt	O	Ethernet Interrupt

Table 4. AXIL Ethernet Interface

Design Flow

IP Customization and Generation

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

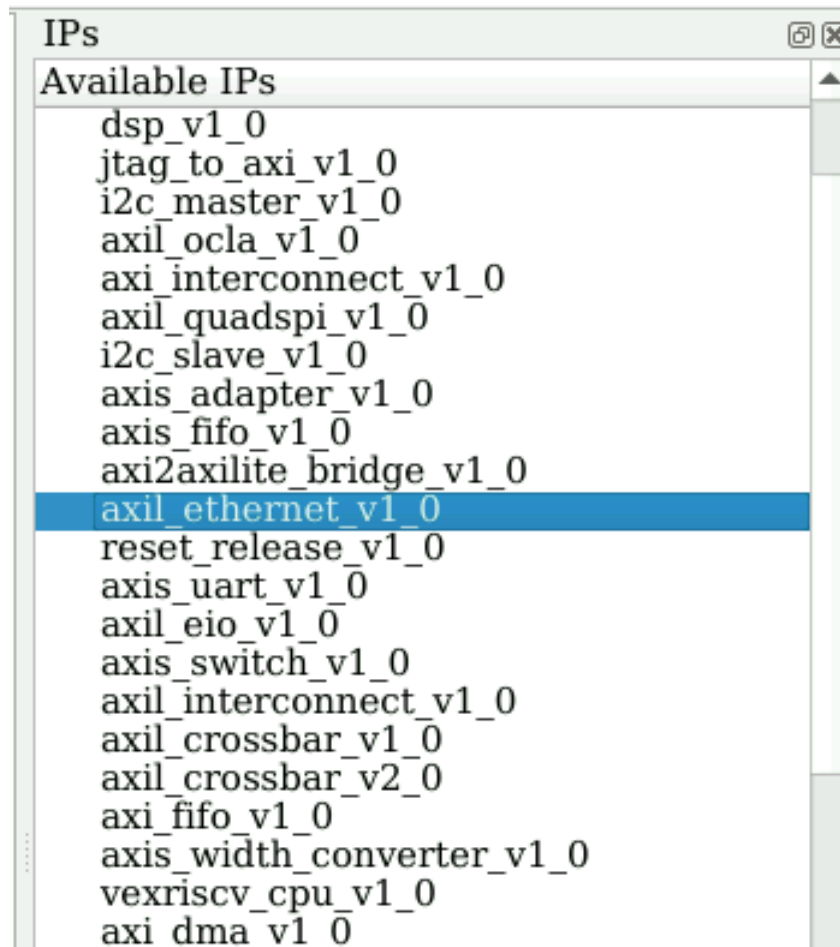
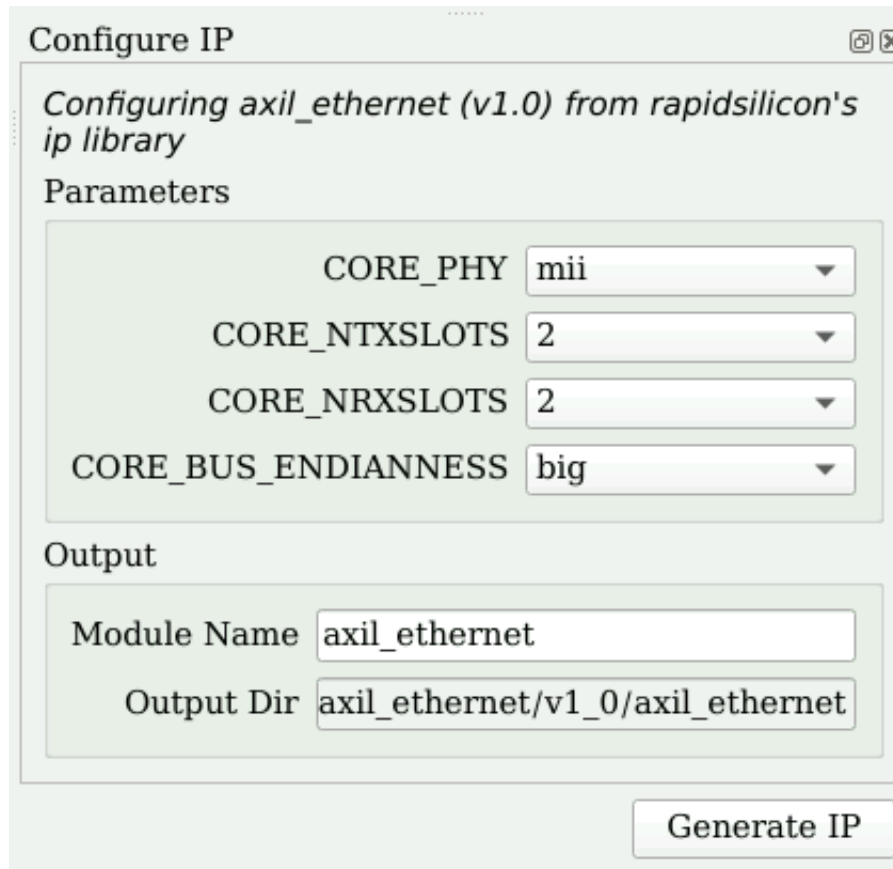


Figure 3. IP list

Parameters Customization: From the IP configuration window, the parameters of the Switch can be configured and Switch features can be enabled for generating a customized Switch 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 AXIL Ethernet.



Configure IP

Configuring axil_ethernet (v1.0) from rapidsilicon's ip library

Parameters

CORE_PHY

CORE_NTXSLOTS

CORE_NRXSLOTS

CORE_BUS_ENDIANNES

Output

Module Name

Output Dir

Generate IP

Figure 4. IP Configuration

Example Design

Overview

The AXILite Ethernet IP can be used as a standalone module or can be utilized in an SoC based system where it is driven via the SoC core. One such design is bundled with the Raptor Design Suite where this IP is incorporated inside of an SoC and a pre-defined test is run on it. The SoC is generated via LiteX with this module included. A pre-defined test is then run on it where a bunch of data is sent and then received utilizing all the TX and RX slots for a complete coverage of the module. A graphical representation of the system can be referred to in the fig 5.

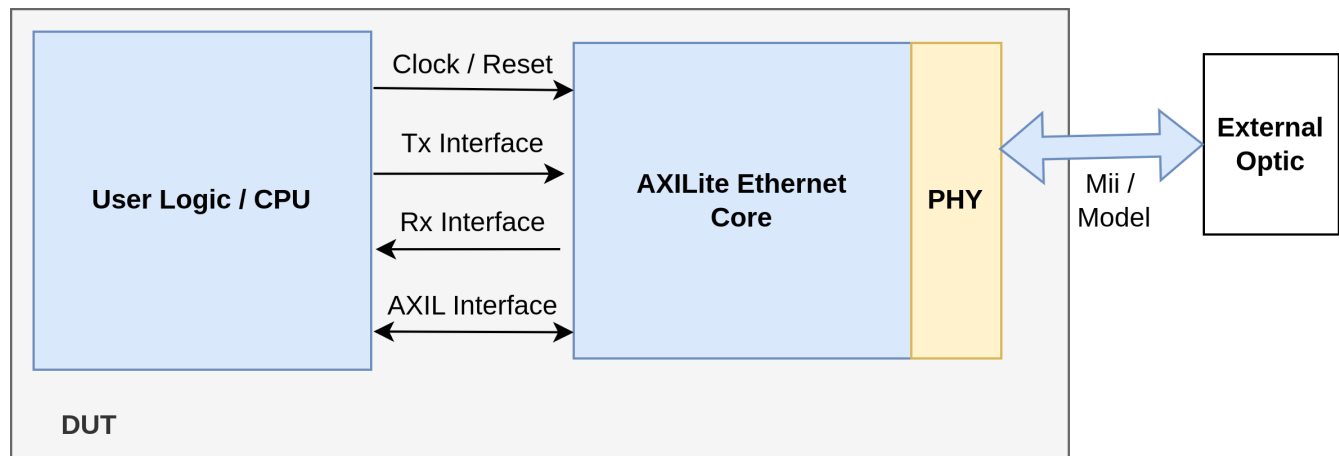


Figure 5. AXIL Ethernet Example Design

Simulating the Example Design

The IP being Verilog HDL, can be simulated via a bunch of industry standard stimulus. For instance, it could be simulated via writing a Verilog Test-bench, or incorporating a soft processor that can stimulate this Ethernet. The bundled example design is stimulated via a LiteX based SoC design where the IP is stimulated by driving it from within the generated LiteX SoC and testing out the data transmission and receiveal.

Synthesis and PnR

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

Test Bench

The Ethernet IP, based Verilog HDL, can be stimulated by any number of industry standard means. These may include simple Verilog test benches or stimulating the Ethernet via some OS or via bare-metal firmwares. The bundled test-bench for this IP is a mixture of Python and C languages making it a bare metal stimulus. Python is used to generate an SoC with this AXIL Ethernet IP attached. After the generation of the SoC, the Ethernet IP is stimulated by providing a bunch of different instructions to the SoC, written in C language. The test bench first defines the addressing of the Ethernet inside the generated SoC. After the addressing, a bunch of different data is sent on all of the TX Slots of the Ethernet. Then the RX Slots are used to receive the same data in a loopback fashion. This test concludes that the data received is the same as the one transmitted, therefore concluding the functionality of this Ethernet IP.

Revision History

Date	Version	Revisions
January 20, 2023	0.01	Initial version AXIL Ethernet User Guide Document