RapidSilicon

# Ethernet MAC
# (Beta Release)

Version 0.1

June 14, 2024

## Copyright



## Trademarks



## Disclaimers

# Contents

# IP Summary

## Introduction

Ethernet MAC IP Core is a pre-verified implements the Media Access Control (MAC) functionality of the Ethernet protocol. It efficiently manages data flow between the FPGA and the network. It ensures access to the shared medium and handling packet reception and transmission. Additionally, it connects the physical network (via GMII/RGMII) to the internal world of the FPGA.

## Features

- Support two physical interfaces.i.e. GMII and RGMII.
- Support FIFO implementation for data buffering.
- Support configurable FIFO depth.
- Supports 10Mbps, 100Mbps and 1000Mbps speed.

# Overview

## Ethernet MAC

An Ethernet MAC IP Core acts as a bridge for enabling communication over Ethernet networks. It implements the Media Access Control(MAC) sublayer of the Ethernet protocol. This core efficiently moves data packets back and forth between the FPGA and the network. Packets leaving the FPGA use the AXI4-Stream TX interface and GMII/RGMII to reach the outside world. Incoming packets travel through the GMII/RGMII and AXI4-Stream RX interface before reaching the FPGA's internal world. This two-way flow management empowers FPGAs to actively participate in network communication.

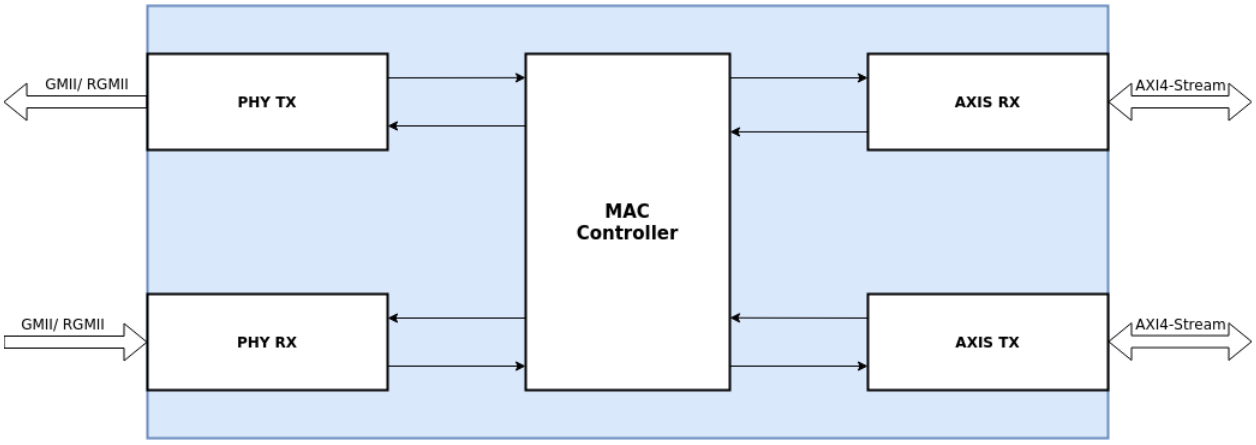


Figure 1: Block Diagram

# IP Specification

Ethernet MAC IP Core offers flexibility for network integration. It supports both GMII and space-saving RGMII interfaces, catering to different board space constraints. Users may tailor data buffering by including or omitting a FIFO and configuring its depth. It adapts to diverse network environments with support for 10Mbps, 100Mbps, and 1Gbps speeds.

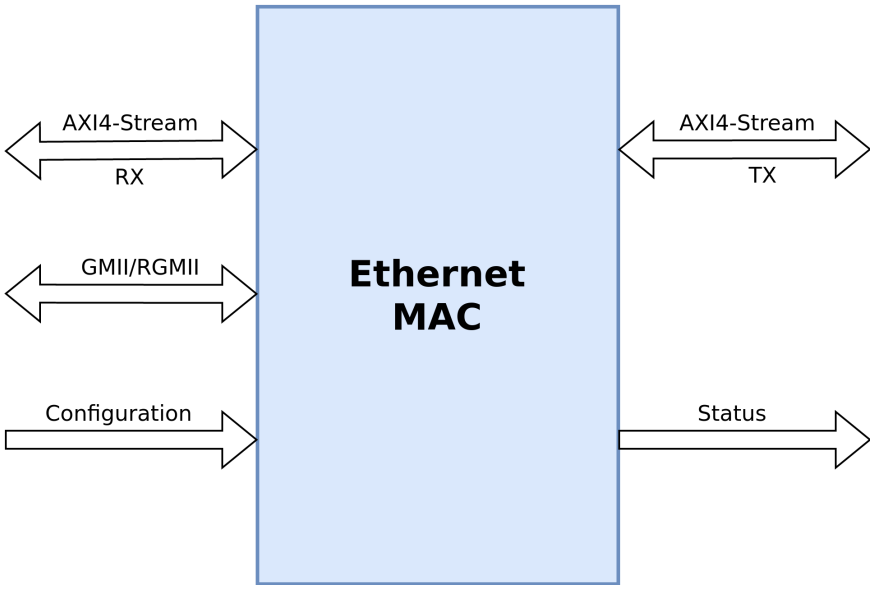


Figure 2: Top Module

## Standards

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

## IP Support Details

The Table 1 gives the support details for Ethernet MAC.

| Compliance | | IP Resources | | | | Tool Flow | | |
|---|---|---|---|---|---|---|---|---|
| **Device** | **Interface** | **Source Files** | **Constraint File** | **Testbench** | **Simulation Model** | **Analyze and Elaboration** | **Simulation** | **Synthesis** |
| VIRGO | AXI4-Stream | 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. Other parameters are kept at their default values.

| **Tool** | Raptor Design Suite | | | |
|---|---|---|---|---|
| **FPGA Device** | VIRGO | | | |
| **Configuration** | | | **Resource Utilization** | |
| Minimum Resource | **Options** | **Configuration** | **Resources** | **Utilized** |
| | INTERFACE | GMII | BRAMS | 0 |
| | FIFO | False | LUTS | 400 |
| | DATA_RATE | 10Mbps | REGISTERS | 224 |
| Maximum Resource | **Options** | **Configuration** | **Resources** | **Utilized** |
| | INTERFACE | RGMII | BRAMS | 3 |
| | FIFO | True | LUTS | 870 |
| | DATA_RATE | 1000Mbps | REGISTERS | 545 |
| | FIFO_DEPTH | 4096 | - | - |

Table 2: Resource Utilization

## Ports

Table 3 lists the top interface ports of the Ethernet MAC.

| Signal Name | Width | I/O | Description |
|---|---|---|---|
| **Clock/Reset** | | | |
| gtx_clk_125MHz | 1 | I | 125MHz Clock signal for the GTX transceiver |
| gtx_clk_12_5MHz | 1 | I | 12.5MHz Clock signal for the GTX transceiver |
| gtx_clk_1_25MHz | 1 | I | 1.25MHz Clock signal for the GTX transceiver |
| gtx_clk90 | 1 | I | 90-degree clock for data transmission in RGMII |
| gtx_rst | 1 | I | Reset signal for the GTX transceiver |
| rx_clk | 1 | I/O | Clock signal for receiving data |
| rx_rst | 1 | I/O | Reset signal for receiving data |
| tx_clk | 1 | I/O | Clock signal for transmitting data |
| tx_rst | 1 | I/O | Reset signal for transmitting data |
| **Transmitter AXI4-Stream** | | | |
| tx_axis_tdata | 8 | I | Parallel data to be transmitted |
| tx_axis_tkeep | 1 | I | Signals which bytes within tx_axis_tdata are valid |
| tx_axis_tvalid | 1 | I | Indicates the presence of valid data in tx_axis_tdata |
| tx_axis_tready | 1 | O | Receiver's readiness to accept data on tx_axis_tdata |
| tx_axis_tlast | 1 | I | Indicates the last data packet in a transmission |
| tx_axis_tuser | 1 | I | Optional user-defined signal for additional information |
| **Receiver AXI4-Stream** | | | |
| rx_axis_tdata | 8 | O | Received parallel data |
| rx_axis_tkeep | 1 | O | Signals which bytes within rx_axis_tdata are valid |
| rx_axis_tvalid | 1 | O | Indicates the presence of valid data in rx_axis_tdata |
| rx_axis_tready | 1 | I | MAC's readiness to accept data on rx_axis_tdata |
| rx_axis_tlast | 1 | O | Indicates the last data packet in a received frame |
| rx_axis_tuser | 1 | O | Optional user-defined signal for additional information |
| **GMII** | | | |
| gmii_rx_clk | 1 | I | Clock signal for receiving data |
| gmii_rxd | 8 | I | Eight data bits received from the PHY |
| gmii_rx_dv | 1 | I | Indicates the validity of received data |
| gmii_rx_er | 1 | I | Indicates an error condition during data reception |
| mii_tx_clk | 1 | I | Clock signal for transmitting data |
| gmii_tx_clk | 1 | O | Clock signal for transmitting data |
| gmii_txd | 8 | O | Eight data bits to be transmitted to the PHY |
| gmii_tx_en | 1 | O | Enables the transmission of data |
| gmii_tx_er | 1 | O | Indicates an error condition during data transmission |
| **RGMII** | | | |
| rgmii_rx_clk | 1 | I | Clock signal for receiving data |
| rgmii_rxd | 4 | I | Four data bits received from the PHY |
| rgmii_rx_ctl | 1 | I | Control signals for receiving data |
| rgmii_tx_clk | 1 | O | Clock signal for transmitting data |

| Signal Name | Width | I/O | Description |
|---|---|---|---|
| rgmii_txd | 4 | O | Four data bits to be transmitted to the PHY |
| rgmii_tx_ctl | 1 | O | Control signals for transmitting data |
| **Status** | | | |
| tx_error_underflow | 1 | O | Error due to data starvation |
| tx_fifo_overflow | 1 | O | Indicates the transmit FIFO is full |
| tx_fifo_bad_frame | 1 | O | Indicates error in transmitted frame |
| tx_fifo_good_frame | 1 | O | Indicates a frame was transmitted without errors |
| rx_fifo_overflow | 1 | O | Indicates FIFO is full |
| rx_fifo_bad_frame | 1 | O | Indicates error in a received frame |
| rx_fifo_good_frame | 1 | O | Indicates a frame was received without errors |
| rx_error_bad_frame | 1 | O | Malformed frame received error |
| rx_error_bad_fcs | 1 | O | Data corruption error during transmission |
| speed | 2 | O | Speed of the Ethernet connection |
| **Configuration** | | | |
| cfg_ifg | 8 | I | Time interval between transmitted Ethernet frames |
| cfg_tx_enable | 1 | I | Enable for transmission of Ethernet frames |
| cfg_rx_enable | 1 | I | Enable for reception of Ethernet frames |

Table 3: Port List

## Parameters

Table 4 lists the parameters of the Ethernet MAC.

| Parameter | Values | Default Value | Description |
|---|---|---|---|
| INTERFACE | GMII, RGMII | GMII | Physical Interface of Ethernet |
| DATA_RATE | 10Mbps, 100Mbps, 1000Mbps | 10Mbps | Speed of Ethernet |
| FIFO | True/ False | False | Ethernet MAC with/without FIFO |
| FIFO_DEPTH | 64, 128, 256, 512, 1024, 2048, 4096 | 4096 | Depth of FIFO for data buffering |

Table 4: Parameters

# Design Flow

## IP Customization and Generation

Ethernet MAC IP core is a part of the Raptor Design Suite Software. Customized IP can be generated from the Raptor's IP configuration window as shown in figure 3.

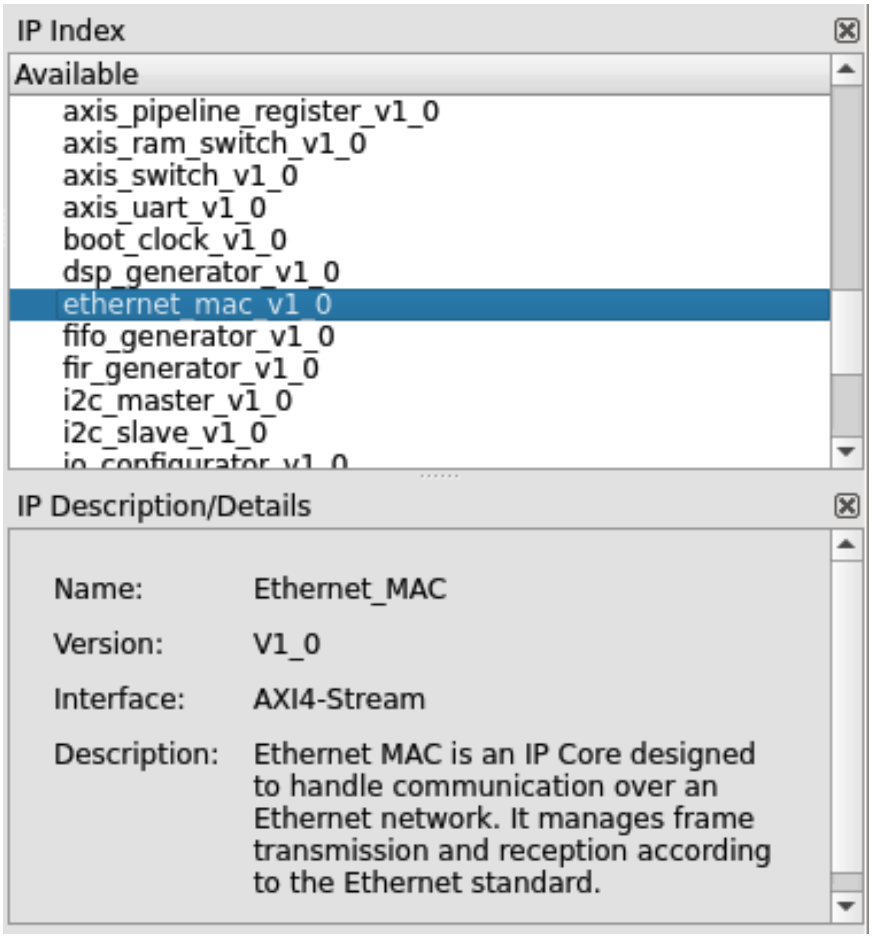


Figure 3: IP List

## Parameters Customization

From the IP configuration window, the parameters of the Ethernet MAC 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. The module name specifies the name of both the Verilog file and the top-level IP name that will be generated based on above configured parameters.

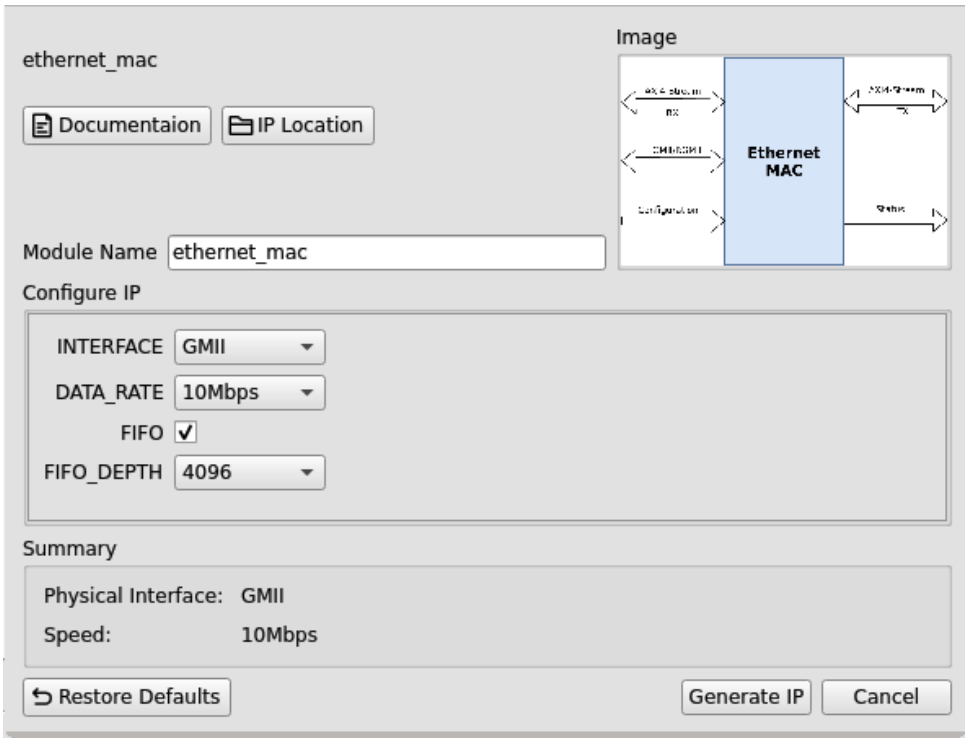


Figure 4: IP Configuration

# Testbench

Ethernet MAC IP Core is provided with a testbench which is based upon Cocotb verification environment. This test injects ethernet packets via the GMII interface. The core receives and verifies the data integrity using Frame Check Sequence (FCS). Then this data will be transmitted over RX AXI4-Stream interface. The core receives the Ethernet packets through TX AXI4-Stream interface and verifies the data integrity using FCS. Then this valid data will be routed to TX GMII physical interface. In this test, sent data and received data is compared to verify the overall functionality of the IP. The test status is shown in Figure 5.

```
**************************************************************************************
** TEST                            STATUS  SIM TIME (ns)  REAL TIME (s)  RATIO (ns/s) **
**************************************************************************************
** test_top.run_test_rx_001        PASS       88020.00          5.04      17450.68  **
** test_top.run_test_rx_002        PASS      837700.00         17.48      47913.32  **
** test_top.run_test_rx_003        PASS     8377800.00        119.72      69977.78  **
** test_top.run_test_tx_001        PASS       88016.00          4.60      19121.77  **
** test_top.run_test_tx_002        PASS      875540.00         17.83      49101.23  **
** test_top.run_test_tx_003        PASS     8756200.00        126.41      69267.81  **
** test_top.run_test_tx_underrun_001 PASS       2800.00          0.14      19664.18  **
** test_top.run_test_tx_underrun_002 PASS      23380.00          0.49      48020.14  **
** test_top.run_test_tx_underrun_003 PASS     234600.00          3.55      66117.26  **
** test_top.run_test_tx_error_001  PASS       2800.00          0.14      19638.72  **
** test_top.run_test_tx_error_002  PASS      23380.00          0.52      44549.19  **
** test_top.run_test_tx_error_003  PASS     234600.00          3.47      67647.21  **
**************************************************************************************
** TESTS=12 PASS=12 FAIL=0 SKIP=0          19544836.01        299.92      65165.75  **
**************************************************************************************
```

Figure 5: Simulation Results

## Simulate IP

For simulation, right click on generated IP Instance and then click "Simulate IP" as shown in Figure 6.

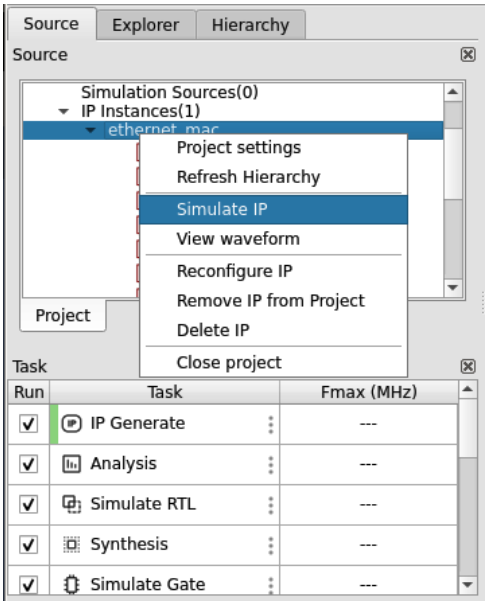


Figure 6: Simulate IP

## Waveform

To view waveform, right click on generated IP Instance and then click "View waveform" as shown in Figure 7.

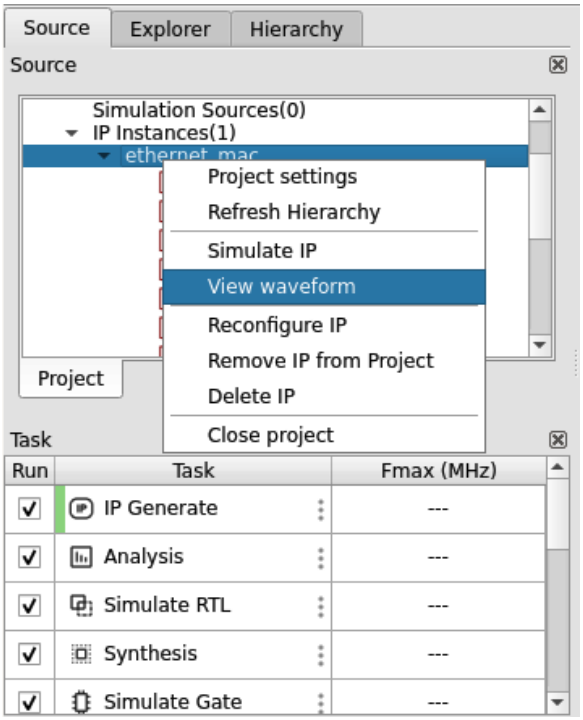


Figure 7: View Waveform

# Revision History

| Date | Version | Revisions |
|---|---|---|
| June 14, 2024 | 0.1 | Initial version Ethernet MAC User Guide |