MII to RMII v2.0

LogiCORE IP Product Guide

Vivado Design Suite

PG146 April 5, 2017

XILINX
ALL PROGRAMMABLE™

Table of Contents

IP Facts

Chapter 1: Overview

Functional Description 5
Licensing and Ordering Information 6

Chapter 2: Product Specification

Standards 7
Performance 7
Resource Utilization 7
Port Descriptions 8

Chapter 3: Designing with the Core

Clocking 10
Resets 10
Protocol Description 10

Chapter 4: Design Flow Steps

Customizing and Generating the Core 14
Constraining the Core 16
Simulation 17
Synthesis and Implementation 17

Chapter 5: Example Design

Implementing the Example Design 19
Simulating the Example Design 20

Chapter 6: Test Bench

Appendix A: Verification, Compliance, and Interoperability

Simulation 23

Appendix B: Migrating and Upgrading

Migrating to the Vivado Design Suite 24
Upgrading in the Vivado Design Suite 24

Appendix C: Debugging

Finding Help on Xilinx.com 25
Debug Tools 27

Appendix D: Additional Resources and Legal Notices

Xilinx Resources 28
References 28
Revision History 29
Please Read: Important Legal Notices 29

Introduction

The Xilinx® LogiCORE™ IP Media Independent Interface (MII) to Reduced Media Independent Interface (RMII) design provides the interface between RMII-compliant ethernet physical media devices (PHY) and Xilinx 10/100 Mb/s ethernet cores such as the AXI Ethernet, AXI Ethernet Lite and Zynq®-7000 PS MAC. These cores provide the traditional MII that requires 16 signals to communicate with an ethernet PHY. The MII to RMII core accepts the 16 signal MII interface and provides a six or seven signal interface to a RMII compliant PHY. Additionally, a fixed 50 MHz reference clock synchronizes the MII to RMII core with both interfaces. The 50 MHz reference clock can be provided by a source external to the host FPGA, or generated within the host FPGA. The MII to RMII core follows the specification defined by the RMII Consortium (version 1.0).

Features

- Core design is based on RMII specification 1.0
- Parameter to allow automatic detection of receive throughput (transmit side always fixed throughput)
- Parameter to select fixed throughput of 10 or 100 Mbits per second
- Fixed clock frequency of 50 MHz

LogiCORE IP Facts Table	
Core Specifics	
Supported Device Family(1)	UltraScale+™ Families, UltraScale™ Architecture, Zynq®-7000 All Programmable SoC, 7 Series
Supported User Interfaces	N/A
Resources	See Table 2-1.
Provided with Core	
Design Files	VHDL
Example Design	VHDL
Test Bench	VHDL
Constraints File	XDC
Simulation Model	Not Provided
Supported S/W Driver	N/A
Tested Design Flows(2)	
Design Entry	Vivado® Design Suite Vivado
Simulation	For supported simulators, see the Xilinx Design Tools: Release Notes Guide.
Synthesis	Vivado Synthesis
Support	
Provided by Xilinx at the Xilinx Support web page	

Notes:

1. For a complete list of supported derivative devices, see the Vivado IP catalog.
2. For the supported versions of the tools, see the Xilinx Design Tools: Release Notes Guide.

Overview

Functional Description

The block diagram for the MII to RMII core is shown in Figure 1-1. This diagram shows the usage, location, and connections for the MII to RMII core.

IMPORTANT: *The same `Ref_Clk` must be provided to clock the MII to RMII core internal to the host FPGA and external to clock the PHY. The clock can be generated either external to, or within the host FPGA, as long as provisions are made to clock all of the necessary blocks.*

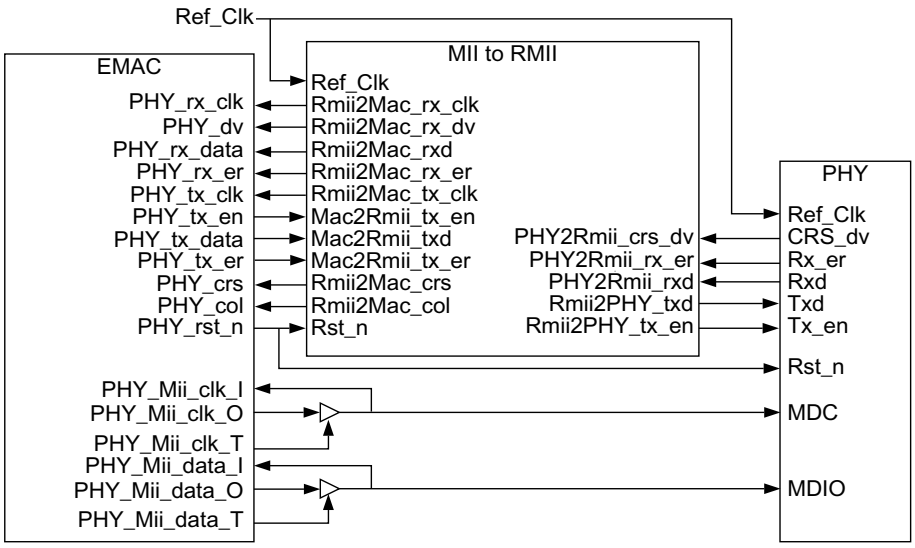

Figure 1-1: **MII to RMII Core Usage Block Diagram**

Licensing and Ordering Information

This Xilinx® LogiCORE IP module is provided at no additional cost with the Xilinx Vivado® Design Suite under the terms of the Xilinx End User License.

Information about this and other Xilinx LogiCORE IP modules is available at the Xilinx Intellectual Property page. For information on pricing and availability of other Xilinx LogiCORE IP modules and tools, contact your local Xilinx sales representative.

Product Specification

Standards

The MII to RMII core follows the specification defined by the RMII Consortium (version 1.0).

Performance

Performance characterization of this core has been done using margin system methodology. The details of the margin system characterization methodology is described in the *Vivado Design Suite User Guide: Designing With IP* (UG896) [Ref 2].

Maximum Frequencies

The maximum frequency achieved on standalone runs is 200 MHz for Virtex®-7/Kintex®-7 families and 150 MHz for Artix®-7 families.

Throughput

No information currently provided for this core.

Resource Utilization

Note: Resources numbers for UltraScale™ architecture and Zynq®-7000 devices are expected to be similar to 7 series device numbers.

7 Series FPGAs

Table 2-1 provides approximate resource counts for the various core options using 7 series devices.

Table 2-1: **Device Utilization – 7 Series FPGAs**

Parameter Values		Device Resources		Performance
Fixed Speed	**Speed 100**	**Slice LUTs**	**Slice Registers**	**F_{Max} (MHz)**
0	0	86	126	200
0	1	75	123	200
1	0	46	98	200
1	1	35	76	200

Port Descriptions

Figure 2-1 shows the ports and interfaces for the MII to RMII IP core and Table 2-2 lists and describes the I/O signals.

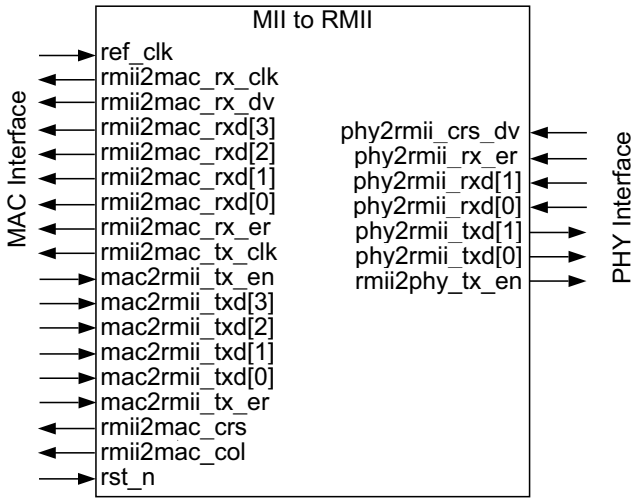

Figure 2-1: **MII to RMII Core Ports and Interfaces**

Table 2-2: **MII to RMII Core I/O Signal Descriptions**

Port	Signal Name	Interface	I/O	Description
RMII PHY Interface Signals				
P1	phy2rmii_crs_dv	PHY	I	Carrier Sense/Data Valid
P2	phy2rmii_rx_er	PHY	I	Receive Error (optional)
P3	phy2rmii_rxd	PHY	I	Receive Data, 2 bits
P4	rmii2phy_txd	PHY	O	Transmit Data, 2 bits

Table 2-2: **MII to RMII Core I/O Signal Descriptions *(Cont'd)***

Port	Signal Name	Interface	I/O	Description
P5	rmii2phy_tx_en	PHY	O	Transmit Enable
Ethernet MAC Interface Signals				
P6	mac2rmii_tx_en	MAC	I	Transmit Enable
P7	mac2rmii_txd	MAC	I	Transmit Data, 4 bits
P8	mac2rmii_tx_er	MAC	I	Transmit Error
P9	rmii2mac_tx_clk	MAC	O	Transmit Clock
P10	rmii2mac_rx_clk	MAC	O	Receive Clock
P11	rmii2mac_col	MAC	O	Ethernet Collision
P12	rmii2mac_crs	MAC	O	Ethernet Carrier Sense
P13	rmii2mac_rx_dv	MAC	O	Receive Data Valid
P14	rmii2mac_rx_er	MAC	O	Receive Error
P15	rmii2mac_rxd	MAC	O	Receive Data, 4 bits
System or MAC Interface Signals				
P16	rst_n	System or MAC	I	MII to RMII core reset (This reset should be connected to the PHY reset signal, if provided by the Ethernet MAC.)
System Interface Signals				
P17	ref_clk	System	I	MII to RMII core and interface clock, 50 MHz

Designing with the Core

This chapter includes guidelines and additional information to facilitate designing with the core.

Clocking

The MII to RMII operates on `ref_clk` clock. This clock is a fixed frequency and has to be equal to 50 MHz.

Resets

The `rst_n` is an active-Low reset to the core.

Protocol Description

The MII to RMII core follows the specification defined by the RMII Consortium (version 1.0).

The diagrams in this section illustrate various signal protocols for the MII to RMII core. The protocols vary from transmit to receive, RMII to PHY, PHY to RMII, MII to RMII, RMII to MII, and data rates of 10 or 100 Mb/s (megabits per second).

Figure 3-1 shows a typical transaction between the physical interface, PHY, and RMII interface on the receive side operating at 100 Mb/s. Several idle dibits, 2 bits at time, can follow the assertion of `Phy2Rmii_crs_dv` and precede the preamble dibits. In this case, the `Phy2Rmii_crs_dv` remains asserted until the final packet dibit.

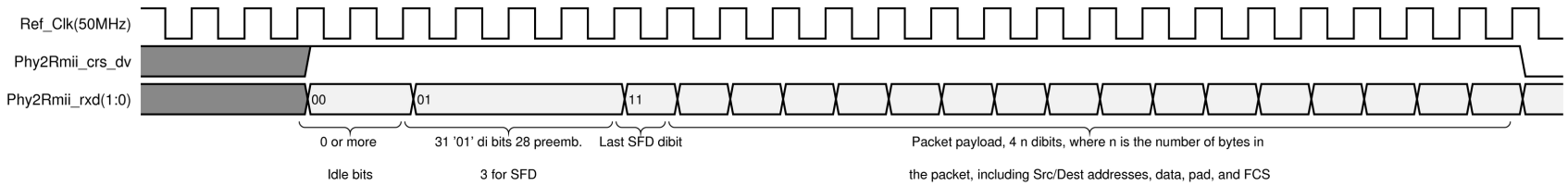

Figure 3-1: **Typical PHY to RMII Receive Transaction at 100 Mb/s**

Figure 3-2 shows another 100 Mb/s PHY to RMII receive transaction. However, in this case `Phy2Rmii_crs_dv` deasserts on the boundary of the last nibble, where two dibits make a nibble and two nibbles make a byte. This indicates the PHY has lost the carrier but still has accumulated packet nibbles to transfer. When this occurs, the PHY cycles `Phy2Rmii_crs_dv` at 25 MHz until the final packet dibit. `Phy2Rmii_crs_dv` deasserts only on nibble boundaries.

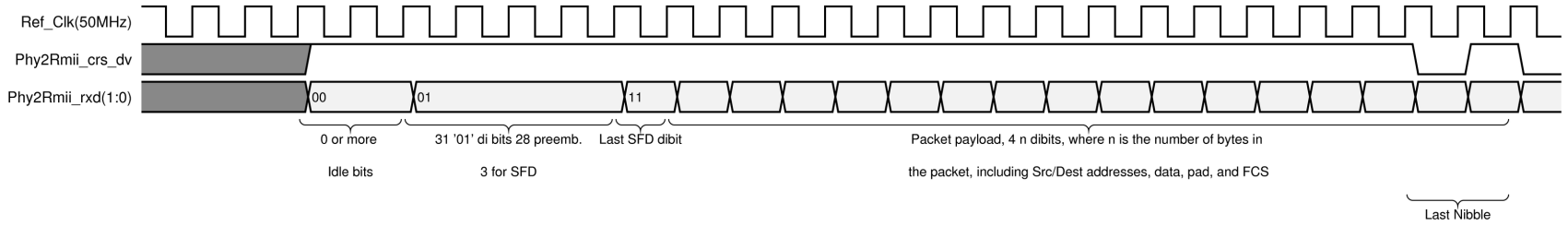

Figure 3-2: **PHY to RMII Receive Transaction at 100 Mb/s, CRS_DV Toggles**

The 10 Mb/s PHY to RMII receive transaction is similar to that of 100 Mb/s transaction except that dibit is asserted by PHY for 10 `Ref_Clk` clock periods.

For 10 Mb/s transfers, the PHY cycles `Phy2Rmii_crs_dv` at 2.5 MHz until the final packet dibit. Again, the signal `Phy2Rmii_crs_dv` deasserts only on nibble boundaries.

Figure 3-3 shows a partial RMII to MII 100 Mb/s receive transaction. The `Rmii2Mac_rx_clk` toggles at 25 MHz with `Rmiii2Mac_rx_dv` and `Rmii2Mac_rxd` transitioning on the falling edge of `Rmii2Mac_rx_clk` to provide amble setup and hold times. When `Rmii2Mac_rx_dv` is deasserted, `Rmii2Mac_rxd` presents 0x0000, idle, to the MAC.

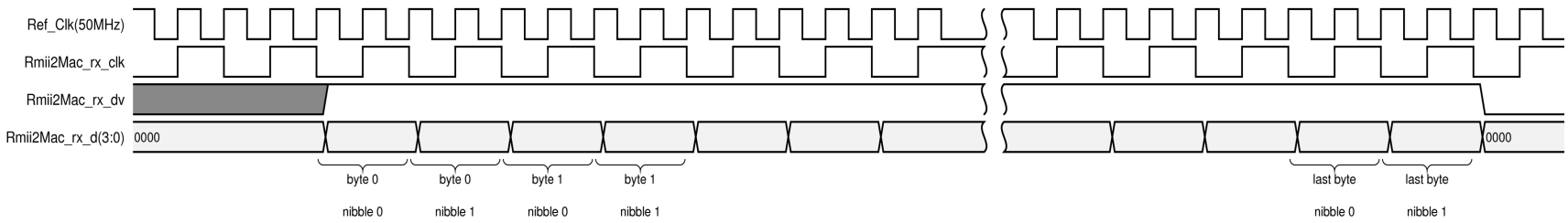

Figure 3-3: **RMII to MII Receive Transaction at 100 Mb/s**

Timing for 10 Mb/s transaction is similar to that of 100 Mb/s except for the `Rmii2Mac_rx_clk` toggles at 2.5 MHz. The timing is 20 periods of `Ref_Clk` for each `Rmii2Mac_rx_clk` period.

Figure 3-4 shows a partial 100 Mb/s MII to RMII transmit transaction. The `Rmii2Mac_tx_clk` toggles at 25 MHz with `Mac2Rmii_tx_en` and `Mac2Rmii_txd` transitioning just after the rising edge of `Rmii2Mac_tx_clk`.

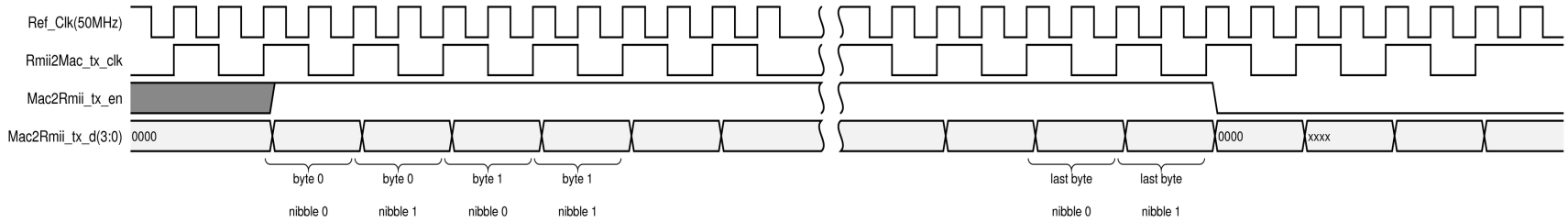

Figure 3-4: **MII to RMII Transmit Transaction at 100 Mb/s**

For 10 Mb/s transaction, the `Rmii2Mac_tx_clk` toggles at 2.5 MHz with `Mac2Rmii_tx_en` and `Mac2Rmii_txd` transitioning just after the rising edge of `Rmii2Mac_tx_clk`.

Figure 3-5 shows a partial 100 Mb/s RMII to PHY transmit transaction. The `Rmii2Mac_tx_clk` toggles at 25 MHz with `Mac2Rmii_tx_en` and `Mac2Rmii_txd` transitioning just after the rising edge of `Rmii2Mac_tx_clk`.

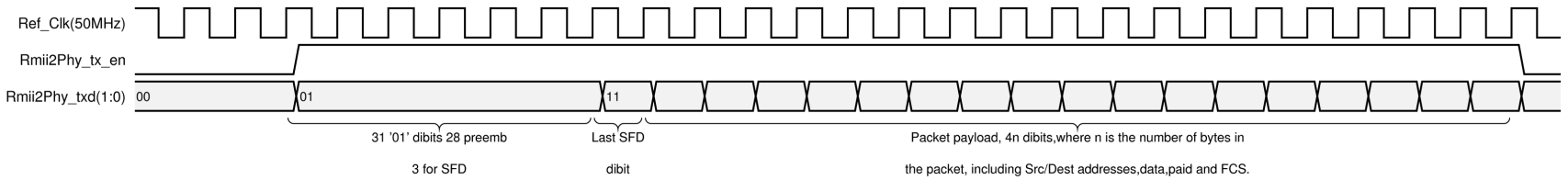

Figure 3-5: **RMII to PHY Transmit Transaction at 100 Mb/s**

For a 10 Mb/s transaction, the `Rmii2Mac_tx_clk` toggles at 2.5 MHz with `Mac2Rmii_tx_en` and `Mac2Rmii_txd` transitioning just after the rising edge of `Rmii2Mac_tx_clk`.

Design Flow Steps

This chapter describes customizing and generating the core, constraining the core, and the simulation, synthesis and implementation steps that are specific to this IP core. More detailed information about the standard Vivado® design flows and the Vivado IP integrator can be found in the following Vivado Design Suite user guides:

- *Vivado Design Suite User Guide: Designing IP Subsystems using IP Integrator* (UG994) [Ref 3]
- *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 2]
- *Vivado Design Suite User Guide: Getting Started* (UG910) [Ref 4]
- *Vivado Design Suite User Guide: Logic Simulation* (UG900) [Ref 5]

Customizing and Generating the Core

This section includes information about using Xilinx® tools to customize and generate the core in the Vivado Design Suite.

If you are customizing and generating the core in the IP integrator, see the *Vivado Design Suite User Guide: Designing IP Subsystems using IP Integrator* (UG994) [Ref 3] for detailed information. IP integrator might auto-compute certain configuration values when validating or generating the design. To check whether the values change, see the description of the parameter in this chapter. To view the parameter value, run the `validate_bd_design` command in the Tcl console.

Vivado Integrated Design Environment

You can customize the IP for use in your design by specifying values for the various parameters associated with the IP core using the following steps:

1. Select the IP from the IP catalog.
2. Double-click the selected IP or select the Customize IP command from the toolbar or right-click menu.

For details, see the *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 2] and the *Vivado Design Suite User Guide: Getting Started* (UG910) [Ref 4].

Note: Figure in this chapter is an illustration of the Vivado IDE. This layout might vary from the current version.

Figure 4-1 shows the Customize IP window settings for MII to RMII IP core.

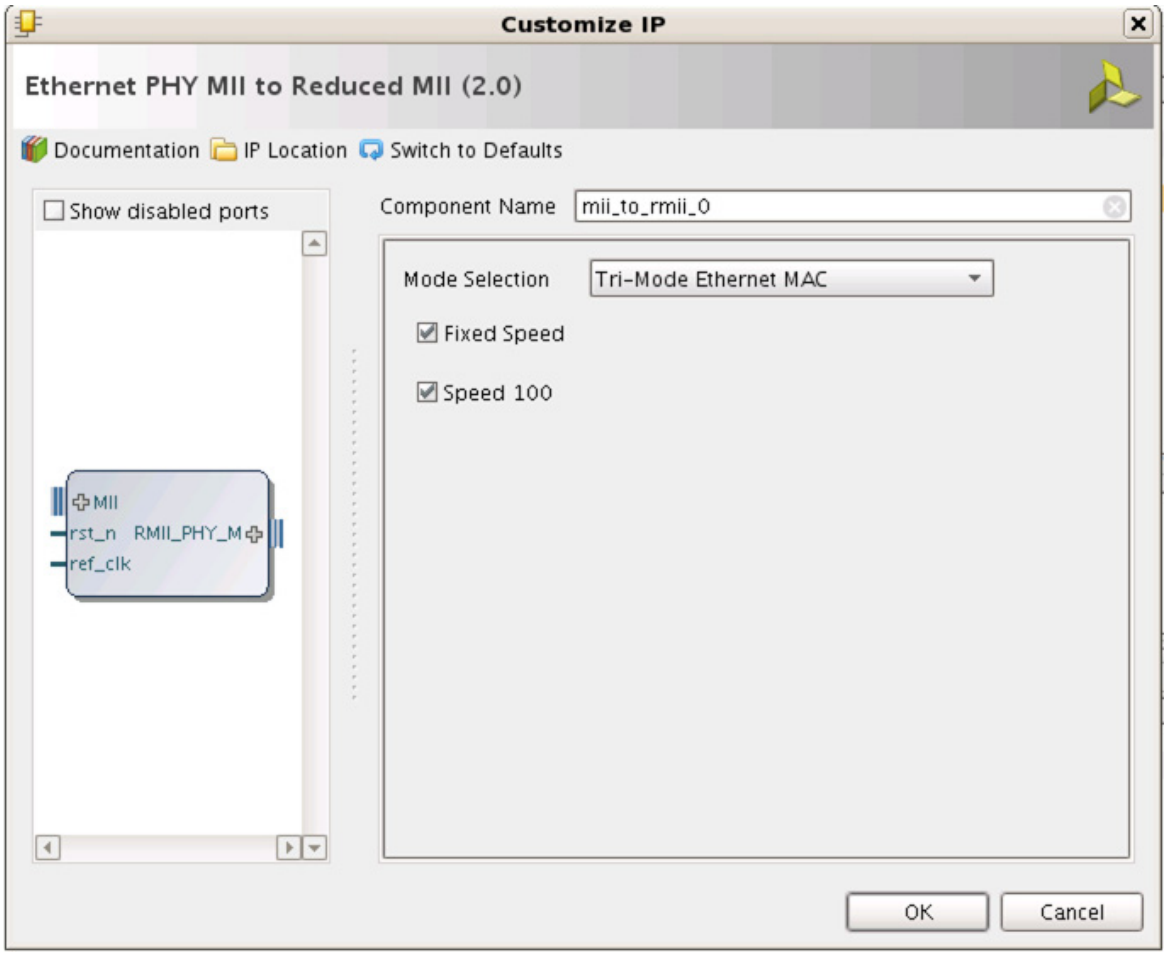

Figure 4-1: **Vivado Customize IP Dialog Box**

- **Component Name** – The base name of the output files generated for the core. Names must begin with a letter and can be composed of any of the following characters: a to z, 0 to 9, and "_".
- **Mode Selection** – Select the interface type on MII side. Tri-Mode Ethernet MAC or Zynq®-7000 AP SoC Gigabit Ethernet Controller.

  Note: Tri-Mode Ethernet MAC is used for AXI Ethernet or AXI Ethernet Lite MAC cores. When using with the AXI Ethernet core, you need to ensure that the IOB properties on the IO ports are disabled in the xdc file generated by the AXI Ethernet core.
- **Fixed Speed** – Fixed ethernet throughput.
- **Speed 100** – When selected speed is 100 Mb/s and when it is not selected, speed is 10 Mb/s.

Note: When **Fixed Speed** is not set, only the receive side automatically detects ethernet throughput. The transmit side operates at a fixed throughput that is determined by **Speed 100**. When **Fixed Speed** is set, **Speed 100** determines the fixed throughput for both transmit and receive.

Output Generation

For details, see the *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 2].

Constraining the Core

This section contains information about constraining the core in the Vivado Design Suite.

Required Constraints

See the *ISE® to Vivado Design Suite Migration Guide* (UG911) [Ref 6] for converting UCF constraints to XDC constraints.

The MII to RMII core requires design constraints to guarantee performance. These constraints should be placed in a XDC for the top-level of the design. The example of the constraint text shown in the example is based on the port names of the MII to RMII core. If these ports are mapped to FPGA pin names that are different, the FPGA pin names should be substituted for the port names in the following example. The constraints listed assume that the core is directly interfaced with the external Ethernet core and that the pins listed in the example actually appear on the FPGA IOBs.

Example 1:

```
create_clock -name Ref_Clk -period 20 -waveform {0 6} [get_ports Ref_Clk]
create_clock -name Rmii2Mac_tx_clk -period 20 -waveform {0 6} [get_ports
Rmii2Mac_tx_clk]
create_clock -name Rmii2Mac_rx_clk -period 20 -waveform {0 6} [get_ports
Rmii2Mac_rx_clk]
set_property IOB TRUE [get_ports Phy2Rmii_rxd[1]]
set_property IOB TRUE [get_ports Phy2Rmii_rxd[0]]
set_property IOB TRUE [get_ports Phy2Rmii_crs_dv]
set_property IOB TRUE [get_ports Phy2Rmii_rx_er]
set_property IOB TRUE [get_ports Phy2Rmii_txd]
set_property IOB TRUE [get_ports Phy2Rmii_en]
```

The following example of the constraint text is based on the port names of the MII to RMII core. If the core has an interface with the AXI Ethernet Lite core in the targeted FPGA, the constraints listed in the example should be used.

Example 2:

```
set_property IOB TRUE [get_ports Rmii2Phy_txd]
set_property IOB TRUE [get_ports Rmii2Phy_en]
```

Device, Package, and Speed Grade Selections

This section is not applicable for this IP core.

Clock Frequencies

This section is not applicable for this IP core.

Clock Management

This section is not applicable for this IP core.

Clock Placement

This section is not applicable for this IP core.

Banking

This section is not applicable for this IP core.

Transceiver Placement

This section is not applicable for this IP core.

I/O Standard and Placement

This section is not applicable for this IP core.

Simulation

This section contains information about simulating IP in the Vivado Design Suite. For comprehensive information about Vivado simulation components, as well as information about using supported third-party tools, see the *Vivado Design Suite User Guide: Logic Simulation* (UG900) [Ref 5].

Synthesis and Implementation

This section contains information about synthesis and implementation in the Vivado Design Suite. For details about synthesis and implementation, see the *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 2].

Example Design

This chapter contains information about the example design provided in the Vivado® Design Suite.

The top module instantiates all components of the core and example design that are needed to implement the design in hardware, as shown in Figure 5-1. This includes clock generator, Traffic Generator, and Ethernet Lite IP to generate Ethernet data.

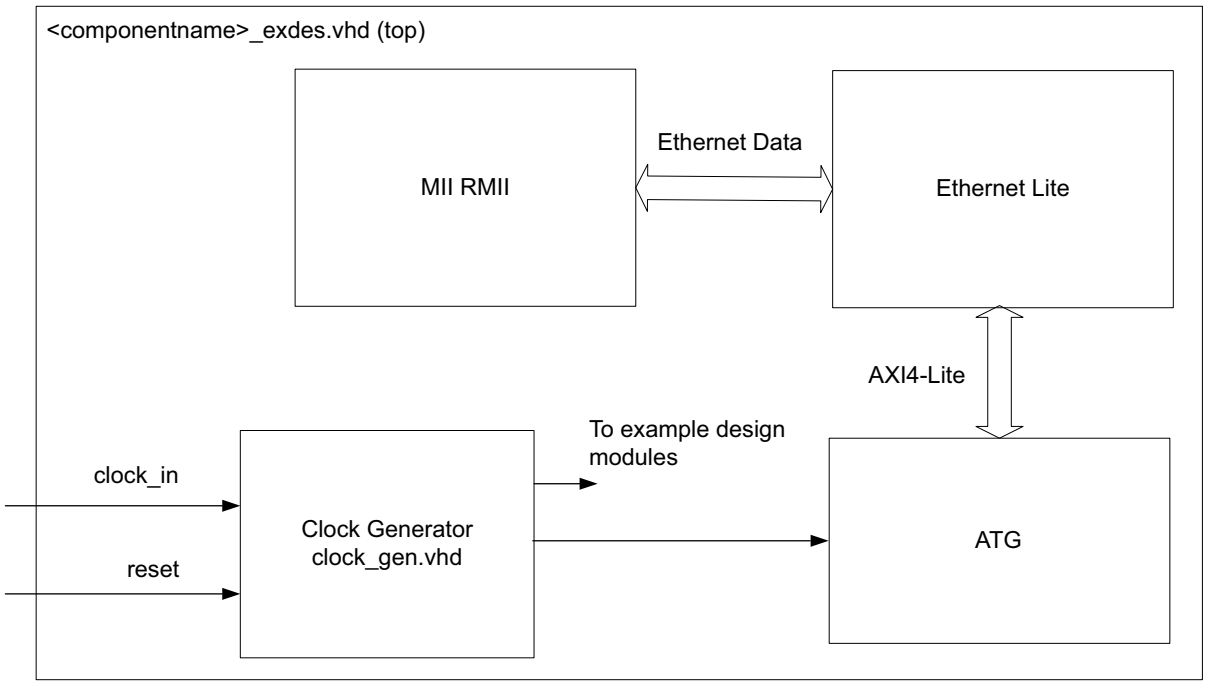

Figure 5-1: **MII to RMII Example Design Block Diagram**

This example design demonstrates transactions of MII to RMII in its different configurations.

- **Clock Generator** – Clocking Wizard IP is used to generate the clock for the example design. It generates clocks for DUT and other modules used in the example design. The example design DUT is kept under reset until Clocking Wizard is locked.
- **AXI Traffic Generator** – The ATG IP is used to program the EthernetLite IP to generate the Ethernet data.
- **AXI EthernetLite** – The AXI EthernetLite IP is used to generate the Ethernet data and is interfaced to MII2RMII IP.

Implementing the Example Design

After following the steps described in Customizing and Generating the Core, page 14 to generate the core, implement the example design as follows:

1. Right-click the core in the Hierarchy window, and select **Open IP Example Design**.
2. A new window pops up, asking you to specify a directory for the example design. Select a new directory or keep the default directory.
3. A new project is automatically created in the selected directory and it is opened in a new Vivado window.
4. In the Flow Navigator (left-side pane), click **Run Implementation** and follow the directions.

Example Design Directory Structure

In the current project directory, a new project with the name `<component_name>_example` is created and the files are generated in the `<component_name>_example/<component_name>_example.srcs/` directory. This directory and its subdirectories contain all the source files that are required to create the MII to RMII example design.

Table 5-1 shows the files delivered in the `<component_name>_example/<component_name>_example.srcs/sources_1/imports/<component_name>/<component_name>/example_design/` directory.

Table 5-1: **Example Design Directory**

Name	Description
<component_name>_exdes.vhd	Top-level HDL file for the example design.

Table 5-2 shows the COE files generated for data transmission.

Table 5-2: **COE Design Directory**

Name	Description
qspi_addr_(1).coe	Delivers address information to the ATG.
qspi_data_(1).coe	Delivers data information to the ATG.
qspi_ctrl_(1).coe	Delivers control information to the ATG.
qspi_mask_(1).coe	Delivers mask information to the ATG.

1. Range from 1 to 3 as each file corresponds to a particular ATG.

Table 5-3 shows the test bench files delivered.

Table 5-3: **Simulation Directory**

Name	Description
<component_name>_exdes_tb.vhd	Test bench for example design.

Table 5-4 shows the `.xdc` files delivered.

Table 5-4: **Constraints Directory**

Name	Description
exdes.xdc	Top-level constraints file for example design.

Example design is verified on the KC705 boards. The board constraints are also specified in `exdes.xdc` file but are commented out. Uncomment the pin constraints while testing on the board.

Simulating the Example Design

Using the MII to RMII example design (delivered as part of the MII to RMII), you can quickly simulate and observe the behavior of the core.

Setting Up the Simulation

The Xilinx® simulation libraries must be mapped into the simulator. If the libraries are not set for your environment, see the *Vivado Design Suite User Guide: Logic Simulation* (UG900) [Ref 5] for assistance compiling Xilinx simulation models and setting up the simulator environment. To switch simulators, click **Simulation Settings** in the Flow Navigator (left pane). In the Simulation options list, change **Target Simulator**.

This section contains instructions for running a functional simulation of the MII to RMII example design. The example design supports functional (behavioral) and post-synthesis simulations.

- To run a functional simulation, click **Run Simulation** in the Flow Navigator (left pane) and then click **Run Behavioral Simulation**.
- To run a post-synthesis simulation, click **Run Simulation** in the Flow Navigator (left pane), and then click **Run Post-Synthesis Functional Simulation**.

Simulation Results

The simulation script compiles the MII to RMII example design and supporting simulation files. It then runs the simulation and checks to ensure that it completed successfully.

If the test passes, then the following message is displayed:

```
Test Completed Successfully
```

If the test fails or does not complete, then the following message is displayed:

```
Test Failed!! Test Timed Out.
```

Test Bench

This chapter contains information about the test bench provided in the Vivado® Design Suite.

Figure 6-1 shows the test bench for the MII to RMII example design. The top-level test bench generates 100 MHz clock and drives initial reset to the example design.

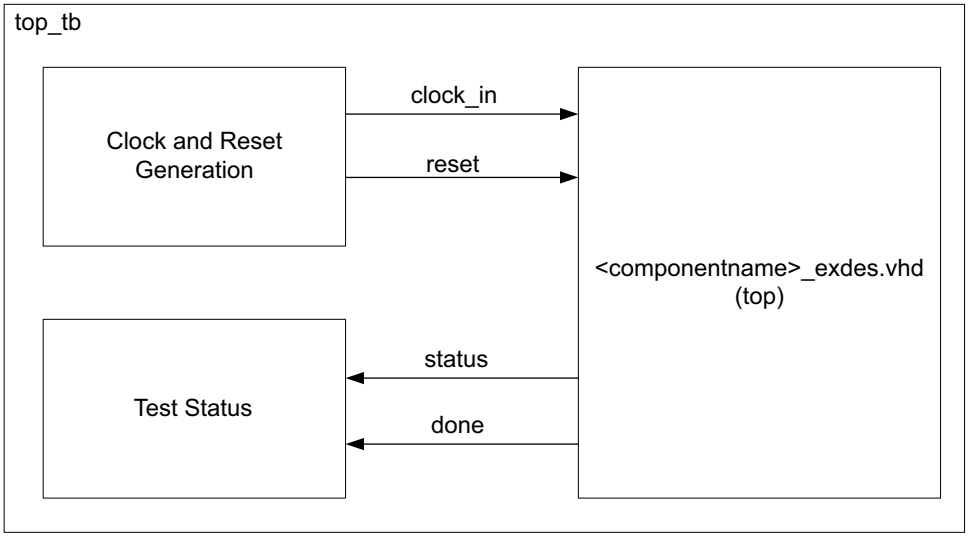

Figure 6-1: **MII to RMII Example Design Test Bench**

Verification, Compliance, and Interoperability

Simulation

The MII to RMII core has been verified through extensive simulation. VHDL test benches were created to simulate functionality and compare operation to the requirements defined by the RMII Consortium Specification.

Migrating and Upgrading

This appendix contains information about migrating a design from ISE® to the Vivado® Design Suite, and for upgrading to a more recent version of the IP core. For customers upgrading in the Vivado Design Suite, important details (where applicable) about any port changes and other impact to user logic are included.

Migrating to the Vivado Design Suite

For information on migrating to the Vivado Design Suite, see the *ISE to Vivado Design Suite Migration Guide* (UG911) [Ref 6].

Upgrading in the Vivado Design Suite

This section provides information about any changes to the user logic or port designations that take place when you upgrade to a more current version of this IP core in the Vivado Design Suite.

Parameter Changes

Added Mode Selection on the MII side.

Debugging

This appendix includes details about resources available on the Xilinx® Support website and debugging tools.

Finding Help on Xilinx.com

To help in the design and debug process when using the MII to RMII, the Xilinx Support web page contains key resources such as product documentation, release notes, answer records, information about known issues, and links for obtaining further product support.

Documentation

This product guide is the main document associated with the MII to RMII. This guide, along with documentation related to all products that aid in the design process, can be found on the Xilinx Support web page or by using the Xilinx Documentation Navigator.

Download the Xilinx Documentation Navigator from the Downloads page. For more information about this tool and the features available, open the online help after installation.

Answer Records

Answer Records include information about commonly encountered problems, helpful information on how to resolve these problems, and any known issues with a Xilinx product. Answer Records are created and maintained daily ensuring that users have access to the most accurate information available.

Answer Records for this core can be located by using the Search Support box on the main Xilinx support web page. To maximize your search results, use proper keywords such as:

- Product name
- Tool message(s)
- Summary of the issue encountered

A filter search is available after results are returned to further target the results.

Master Answer Record for the MII to RMII

AR: 54363

Technical Support

Xilinx provides technical support in the Xilinx Support web page for this LogiCORE™ IP product when used as described in the product documentation. Xilinx cannot guarantee timing, functionality, or support if you do any of the following:

- Implement the solution in devices that are not defined in the documentation.
- Customize the solution beyond that allowed in the product documentation.
- Change any section of the design labeled DO NOT MODIFY.

To contact Xilinx Technical Support, navigate to the Xilinx Support web page.

Debug Tools

There are many tools available to address MII to RMII design issues. It is important to know which tools are useful for debugging various situations.

Vivado Design Suite Debug Feature

The Vivado® Design Suite debug feature inserts logic analyzer and virtual I/O cores directly into your design. The debug feature also allows you to set trigger conditions to capture application and integrated block port signals in hardware. Captured signals can then be analyzed. This feature in the Vivado IDE is used for logic debugging and validation of a design running in Xilinx devices.

The Vivado logic analyzer is used with the logic debug IP cores, including:

- ILA 2.0 (and later versions)
- VIO 2.0 (and later versions)

See the *Vivado Design Suite User Guide: Programming and Debugging* (UG908) [Ref 7].

Appendix D

Additional Resources and Legal Notices

Xilinx Resources

For support resources such as Answers, Documentation, Downloads, and Forums, see Xilinx Support.

References

These documents provide supplemental material useful with this product guide:

1. RMII Specification, written by the RMII Consortium Version 1.0
2. *Vivado® Design Suite User Guide: Designing with IP* (UG896)
3. *Vivado Design Suite User Guide: Designing IP Subsystems Using IP Integrator* (UG994)
4. *Vivado Design Suite User Guide: Getting Started* (UG910)
5. *Vivado Design Suite User Guide: Logic Simulation* (UG900)
6. *ISE® to Vivado Design Suite Migration Guide* (UG911)
7. *Vivado Design Suite User Guide: Programming and Debugging* (UG908)
8. *Vivado Design Suite User Guide: Implementation* (UG904)
9. IEEE Std. 802.3

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
04/05/2017	2.0	Updated to recommend the usage of MII to RMII IP with the AXI Ethernet IP.
11/18/2015	2.0	Added support for UltraScale+ families.
04/02/2014	2.0	• Updated Resource Utilization section. • Updated Clocking section. • Updated Example Design and Test Bench chapters.
12/18/2013	2.0	Revision number advanced to 2.0 to align with core version number. • Added UltraScale and IP Integrator support. • Updated description Performance section. • Updated figure and description in Generating the Core chapter. • Updated timing constraints in Constraints chapter. • Added Simulation, Synthesis, Example Design, and Test Bench chapters. • Updated Migrating Appendix.
03/20/2013	1.0	Initial Xilinx release of the product guide and replaces DS476.

Please Read: Important Legal Notices