MII to RMII

DS476 May 5, 2004 **Product Overview**

Introduction

The MII_to_RMII design described in this document provides the Reduced Media Independent Interface between RMII compliant ethernet physical media devices (PHY) and Xilinx 10/100 Mb/s ethernet cores such as the OPB Ethernet, OPB Ethernet Lite, and the PLB Ethernet. These cores provide the traditional Media Independent Interface (MII) that requires sixteen signals to communicate with an ethernet PHY. The MII_to_RMII accepts the sixteen signal MII interface and provides a six or seven signal interface to an RMII compliant PHY. Additionally, a fixed 50 MHz reference clock synchronizes the MII_to_RMII with both interfaces. The 50 MHz reference clock may be provided by a source external to the host FPGA, or generated within the host FPGA. The MII_to_RMII follows the specification defined by the RMII Consortium found on the internet site http://broadband.spirentcom.com/technology/chipsolutions/rmii_1_2.pdf.

Features

The MII_to_RMII is a soft IP core designed for Xilinx FPGAs and contains the following features:

- MII Interface
- RMII Interface
- 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
- A fixed clock frequency of 50 MHz.

MII_to_RMII Parameters

To allow the user to obtain an MII_to_RMII that is uniquely tailored for their system, certain features are parameterizable. This allows the user to have a design that only utilizes the resources required by their system and runs at the best possible performance. The features that are parameterizable in the Xilinx MII_to_RMII are shown in Table 1.

LogiCORE™ Facts		
Core Specifics		
Supported Device Family	Virtex-4™, Virtex-II Pro™, Virtex-II™, Virtex™, Virtex™-E, Spartan™-II, Spartan™-IIE, Spartan™-III	
Version of Core	mmi_to_rmii	v1.00b
Resources Used		
	Min	Max
Core FPGA IOBs	6	8
Total Core I/O	25	25
LUTs	36	112
FFs	59	185
Block RAMs	0	0
Provided with Core		
Documentation	Product Specification	
Design File Formats	VHDL	
Constraints File	N/A	
Verification	N/A	
Instantiation Template	N/A	
Reference Designs	None	
Design Tool Requirements		
Xilinx Implementation Tools	6.2i or later	
Verification	N/A	
Simulation	ModelSim PE/SE 5.7d or later	
Synthesis	XST	
Support		
Support provided by Xilinx, Inc.		

Table 1: **MII_to_RMII Parameters**

Grouping / Number		Feature / Description	Parameter Name	Allowable Values	Default Value	VHDL Type
MII_to_RMII	G1	Fixed ethernet throughput	C_FIXED_SPEED	'0' = Auto (rec. only) '1' = Fixed	'1'	std_logic
	G2	Throughput set at 100 Mbps	C_SPEED_100	'0' = 10 Mb/s '1' = 100 Mb/s	'1'	std_logic

Notes:

1. When C_FIXED_SPEED = '0', only the receive side automatically detects ethernet throughput. The transmit side will operate at a fixed throughput that is determined by C_SPEED_100. When C_FIXED_SPEED is set to '1', C_SPEED_100 determines the fixed throughput for both transmit and receive.

Allowable Parameter Combinations

There are no restrictions on parameter combinations in the Xilinx MII_to_RMII design other than setting the parameters to proper logic values, '0' or '1'. The notes in Table define the interaction of the two parameters.

MII_to_RMII I/O Signals

The I/O signals for the MII_to_RMII are listed in Table 2. The interfaces referenced in this table are shown in Figure 2 in the MII_to_RMII usage diagram.

Table 2: **MII_to_RMII I/O Signals**

Grouping		Signal Name	Interface	I/O	Description
RMII PHY	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
	P5	Rmii2Phy_tx_en	PHY	O	Transmit Enable
Ethernet Mac	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	P16	Rst_n	System or MAC	I	MII_to_RMII reset (OPB Ethernet and PLB_Ethernet provide PHY_Rst_n that may be used while OPB Ethernet Lite does not provide a PHY reset.)
System	P17	Ref_Clk	System	I	MII_to_RMII and interface clock, 50 MHz

Figure 1 shows the ports and interfaces for the MII_to_RMII.

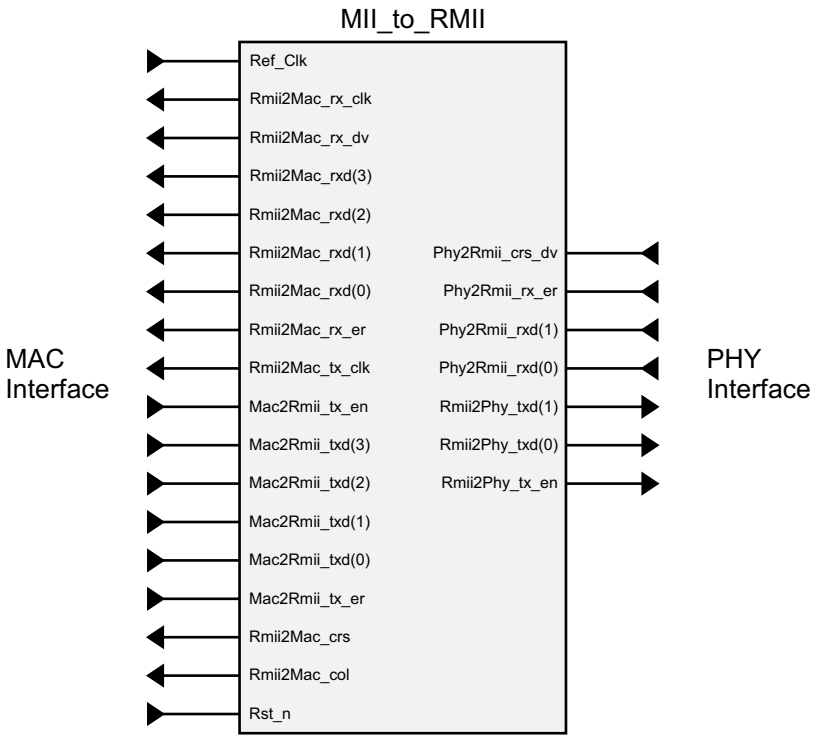

Figure 1: **MII to RMII Ports and Interfaces**

MII_to_RMII Port Dependencies

None.

MII_to_RMII Register Descriptions

None.

MII_to_RMII Interrupt Descriptions

None.

MII_to_RMII Signal Protocols

The following diagrams illustrate various signal protocols for the MII_to_RMII. 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 Mbps (megabits per second). Figure 2 shows a typical transaction between the physical interface, PHY, and RMII interface on the receive side operating at 100 Mbps.

Note that several idle dibits, two bits at time that is, may 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.

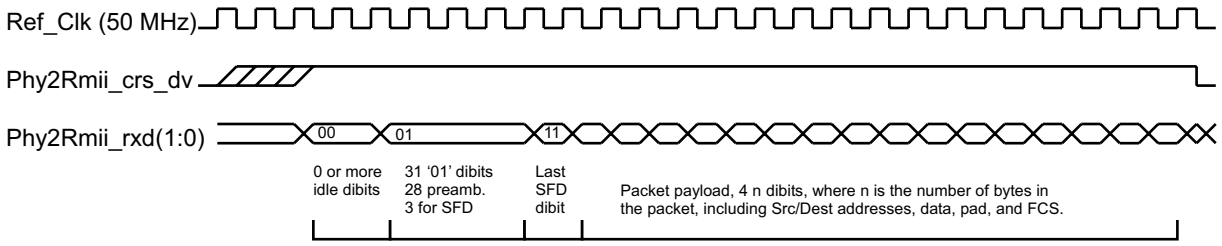

Figure 2: **Typical PHY to RMII Receive Transaction at 100 Mbps**

Figure 3 shows another 100 Mbps 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 will cycle Phy2Rmii_crs_dv at 25 MHz until the final packet dibit. Phy2Rmii_crs_dv deasserts only on nibble boundaries.

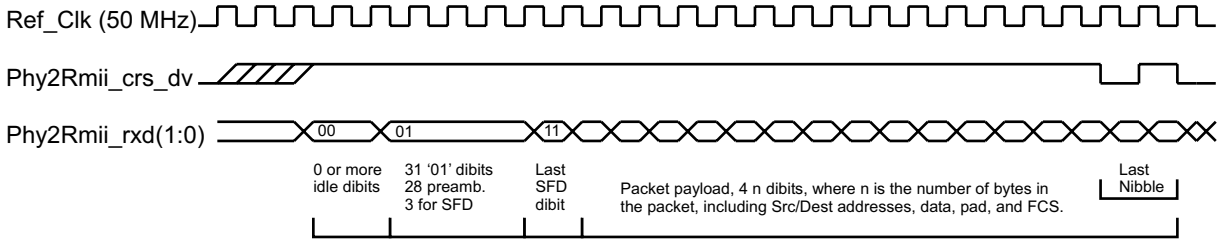

Figure 3: **PHY to RMII Receive Transaction at 100 Mbps, CRS_DV Toggles**

Figure 4 shows a typical 10 Mbps PHY to RMII receive transaction. Each dibit is asserted by the PHY for 10 Ref_Clk clock periods.

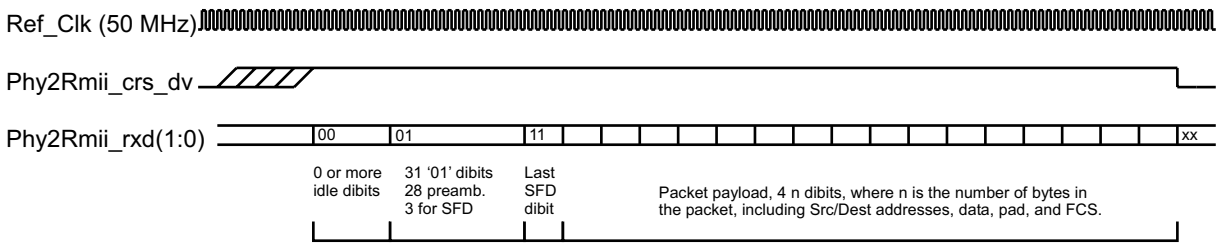

Figure 4: **Typical PHY to RMII Receive Transaction at 10 Mbps**

Figure 5 shows a 10 Mbps PHY to RMII receive transaction. In this case Phy2Rmii_crs_dv deasserts on the boundary of the last nibble. For 10 Mbps transfers, the PHY will cycle Phy2Rmii_crs_dv at 2.5 MHz until the final packet dibit. Again, the signal Phy2Rmii_crs_dv deasserts only on nibble boundaries.

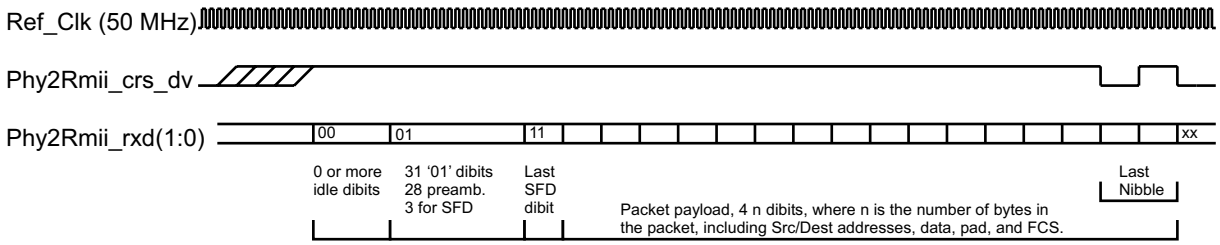

Figure 5: **PHY to RMII Receive Transaction at 10 Mbps, CRS_DV Toggles**

Figure 6 shows a partial 100 Mbps RMII to MII receive transaction. Note that 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 Rmiii2Mac_rx_dv is deasserted, Rmii2Mac_rxd presents 0b0000, idle, to the MAC.

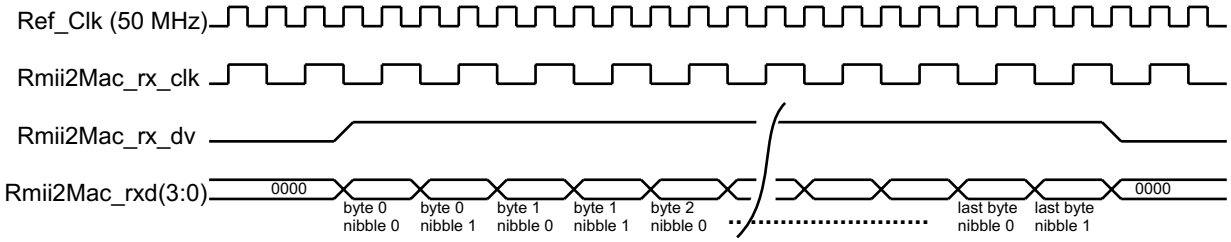

Figure 6: **RMII to MII Receive Transaction at 100 Mbps**

Figure 7 shows a partial 10 Mbps RMII to MII receive transaction. Note that the Rmii2Mac_rx_clk toggles at 2.5 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 Rmiii2Mac_rx_dv is deasserted, Rmii2Mac_rxd presents 0b0000, idle, to the MAC. The timing is 20 periods of Ref_Clk for each Rmii2Mac_rx_clk period.

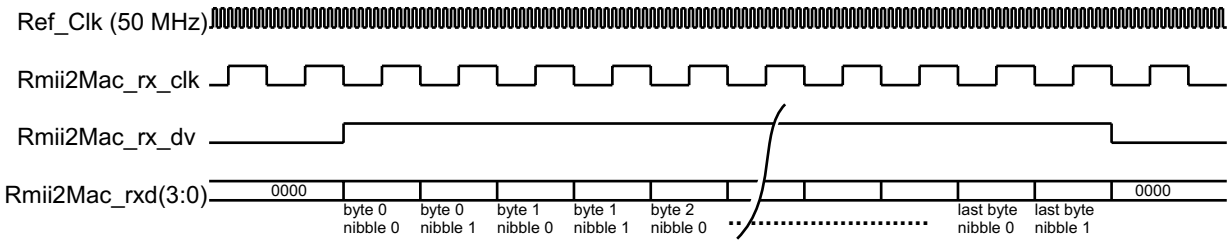

Figure 7: **RMII to MII Receive Transaction at 10 Mbps**

Figure 8 shows a partial 100 Mbps MII to RMII transmit transaction. Note that 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.

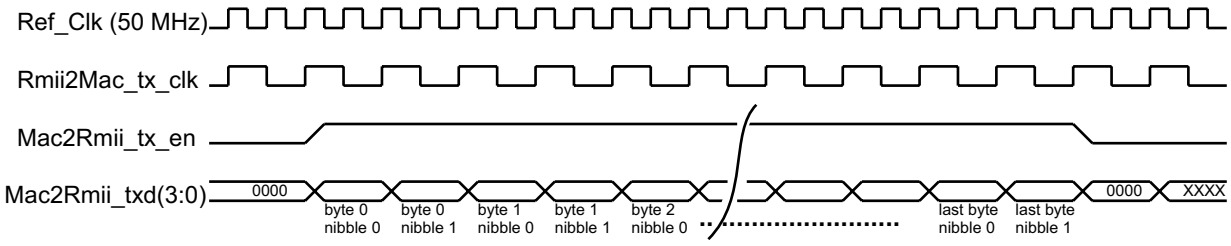

Figure 8: **MII to RMII Transmit Transaction at 100 Mbps**

Figure 9 shows a partial 10 Mbps MII to RMII transmit transaction. Note that 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.

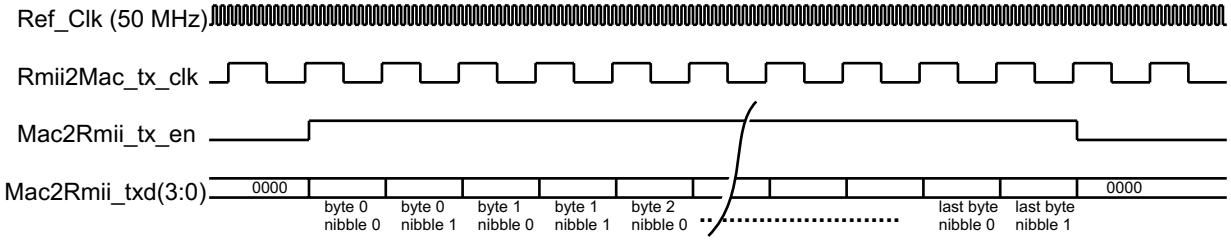

Figure 9: **MII to RMII Transmit Transaction at 10 Mbps**

Figure 10 shows a partial 100 Mbps RMII to PHY transmit transaction. Note that 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.

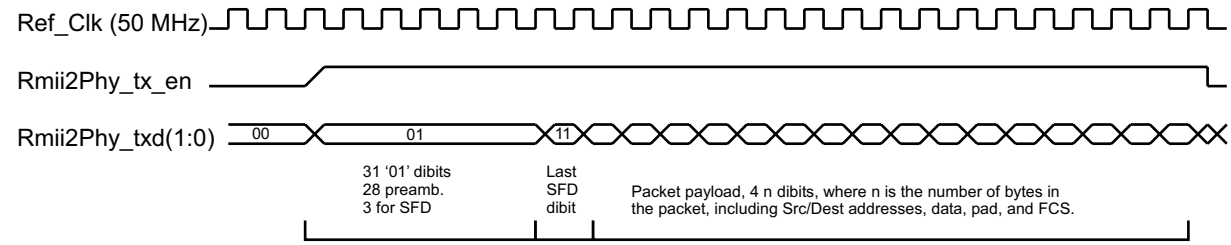

Figure 10: ***R*MII to PHY Transmit Transaction at 100 Mbps**

Figure 11 shows a partial 10 Mbps RMII to PHY transmit transaction. Note that 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.

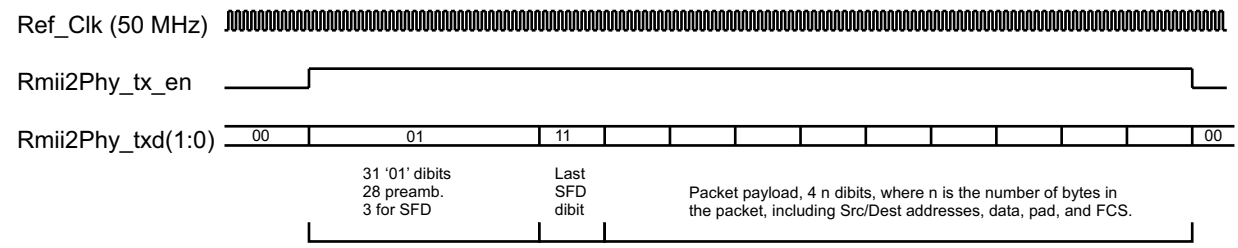

Figure 11: ***R*MII to PHY Transmit Transaction at 10 Mbps**

MII_to_RMII Usage Diagram

The block diagram for the MII_to_RMII is shown in Figure 12. This diagram shows the usage, location, and connections for the MII_to_RMII. The Ref_Clk must be provided to clock the MII_to_RMII internal to the host FPGA and external to clock the PHY. This clock may be generated either external to, or within the host FPGA so long as provisions are made to clock all of the necessary blocks.

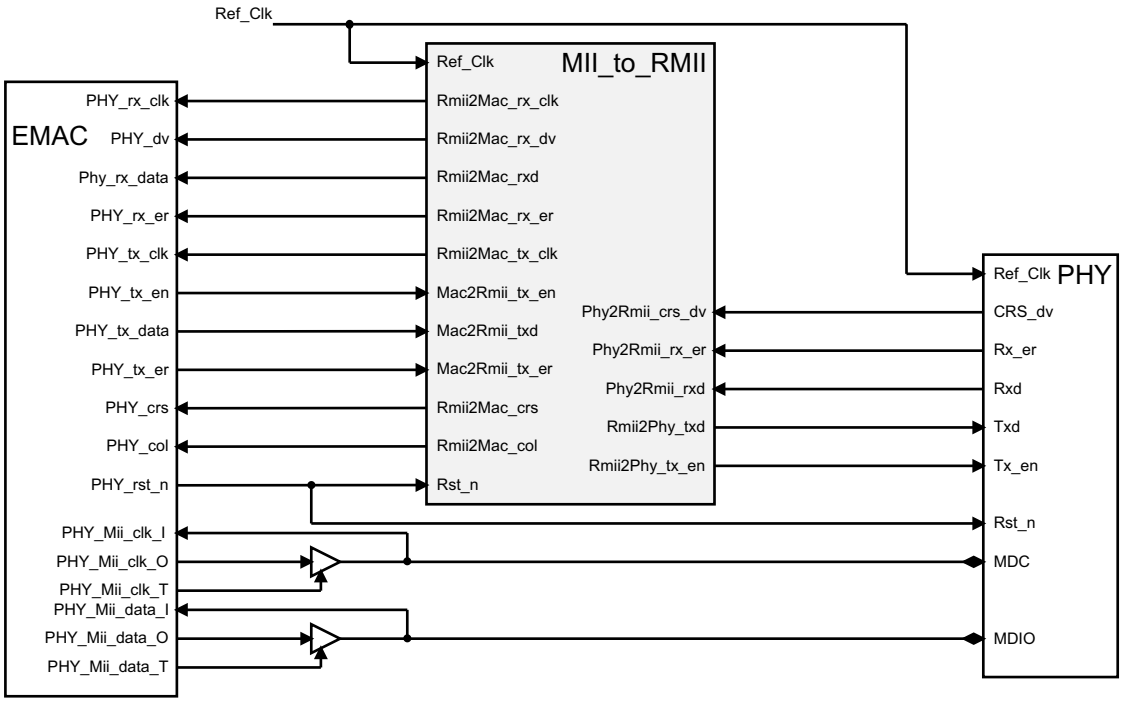

Figure 12: **MII_to_RMII Usage Block Diagram**

Design Implementation

Design Tools

Synplicity's Synplify Pro was the synthesis tool used to synthesize the MII to RMII. The EDIF netlist output from Synplify Pro is then input to the Xilinx Alliance tool suite for actual device implementation.

Design Verification

The MII_to_RMII 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.

Target Technology

The intended target technologies include the full range of Virtex and Spartan FPGA families.

Design Constraints

The MII_to_RMII Core requires design constraints to guarantee performance. These constraints should be placed in a .UCF file for the top level of the design. The following example of the constraint text 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 example below.

```
NET "Ref_Clk" TNM_NET = "Ref_Clk_GRP";
TIMESPEC "TSTXOUT" = FROM "Ref_Clk_GRP" TO "PADS" 5 ns;
TIMESPEC "TSRXIN"  = FROM "PADS" TO "Ref_Clk_GRP" 3 ns;
NET "Ref_Clk"         USELOWSKEWLINES;
NET "Rmii2Mac_tx_clk" USELOWSKEWLINES;
NET "Rmii2Mac_rx_clk" USELOWSKEWLINES;
NET "Ref_Clk"         MAXSKEW= 2.0 ns;
NET "Rmii2Mac_tx_clk" MAXSKEW= 2.0 ns;
NET "Rmii2Mac_rx_clk" MAXSKEW= 2.0 ns;

NET "Ref_Clk"         PERIOD = 20 ns HIGH  6 ns;
NET "Rmii2Mac_tx_clk" PERIOD = 40 ns HIGH 18 ns; # 100 Mb/s
NET "Rmii2Mac_rx_clk" PERIOD = 40 ns HIGH 18 ns; # 100 Mb/s
#NET "Rmii2Mac_tx_clk" PERIOD = 400 ns HIGH 180 ns; # 10 Mb/s
#NET "Rmii2Mac_rx_clk" PERIOD = 400 ns HIGH 180 ns; # 10 Mb/s

NET "Phy2Rmii_rxd<1>" NODELAY;
NET "Phy2Rmii_rxd<0>" NODELAY;
NET "Phy2Rmii_crs_dv" NODELAY;
NET "Phy2Rmii_rx_er"  NODELAY;

INST Phy2Rmii_rxd<1>  IOB = true;
INST Phy2Rmii_rxd<0>  IOB = true;
INST Phy2Rmii_crs_dv  IOB = true;
INST Phy2Rmii_rx_er   IOB = true;
INST Rmii2Phy_txd     IOB = true;
INST Rmii2Phy_tx_en   IOB = true;
```

Device Utilization and Performance Benchmarks

This section will be updated when the design has been completed. It will contain the resources and timing for various values of the parameters.

The MII_to_RMII benchmarks are shown in Table 3 for a Virtex-II -5 FPGA.

Table 3: **MII_to_RMII FPGA Performance and Resource Utilization Benchmarks (Virtex-II -5)**

Parameter Values		Device Resources			f_{MAX}
C_FIXED_SPEED	C_SPEED_100	Slices	Slice Flip-Flops	4-input LUTs	MHz
0	0	141	185	112	> 100
0	1	131	169	108	> 100
1	0	99	148	42	> 100
1	1	59	85	36	> 100

Notes:

1. These benchmark designs contain only the MII_to_RMII with registered inputs/outputs without any additional logic. Benchmark numbers approach the performance ceiling rather than representing performance under typical user conditions.

Specification Exceptions

None.

Reference Documents

The following documents contain reference information important to understanding the MII_to_RMII design:

- IEEE Std. 802.3
- RMII Specification, written by the RMII Consortium

http://broadband.spirentcom.com/technology/chipsolutions/rmii_1_2.pdf.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
05/07/03	1.0	Initial release
07/09/03	1.2	Update for EDK Granite
01/09/04	1.3	Update family per CR 181443; update copyright date
02/06/04	1.4	Added IOB count and protocol wave diagrams
03/05/04	1.5	Added implementation constraints, changed synthesis to XST, changed several Overview items to N/A
03/16/04	1.6	Updated resource use tables.
05/05/04	1.7	Added Virtex-4™ in supported families.