RGMII to AXI Stream IP

Green Electrons GmbH

June 17, 2022

Contents

List of Figures

List of Tables

1 Support and Contact

The original web page for RGMII to AXI Stream can be found here.

Might you have any questions regarding this IP or might you face any difficulties during using the IP, please write an email to mamsadegh@green-electrons.com. We ensure your question or issue will be addressed.

Migh you think anything is missing from the IP, or might you need any further customization to the IP or its related interfaces so that you can use it successfully in your project, please also write mamsadegh@green-electrons.com.

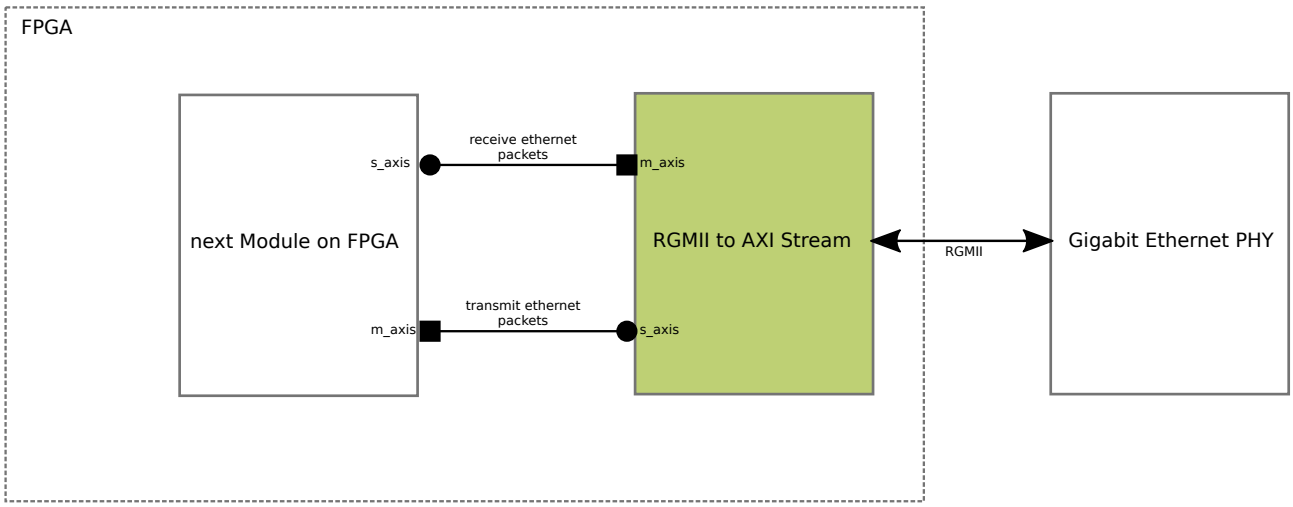

Figure 1: Simplified connectivity of the ip. (RGMII PHY)

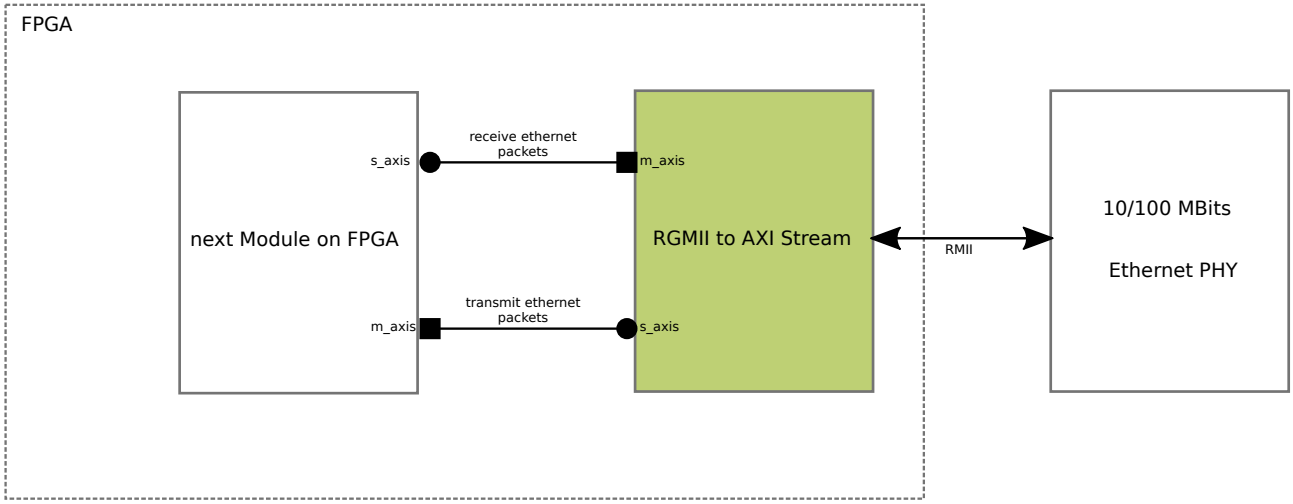

Figure 2: Simplified connectivity of the ip. (RMII PHY)

2 Introduction

RGMII-to-AXIS IP provides an easy to use interface between a RGMII or a RMII PHY and other FPGA modules.

Figure 1 show how the IP gets connected to a RGMII PHY and to other FPGA modules. Figure 2 show how the IP gets connected to a RMII PHY and to other FPGA modules.

Note that in both cases the IP is called *RGMII to AXIS* however it is capable of talking to both RGMII and RMII PHYs.

3 Tools

The IP is originally created and tested with Xilinx 2020.2 tools. The IP can be used with this version or any later version of the Xilinx tool.

Compatibility of the IP is planned to expand to Intel and Microchip FPGAs as well. This is currently an ongoing task.

index	Folder	Description
1	constraints	Timing and pinning constraints files
2	doc	Documentation and design tests snapshots
3	downloads	Key downloaded datasheets and documents
4	packaged_ip	Vivado packaged ip (for RGMII PHY)
5	packaged_ip_rmii	Vivado packaged ip (for RMII PHY)
6	tcl	Vivado tcl scripts
7	verilog	Verilog RTL source code of the IP
8	verilog_sim	Verilog code of simulation and verification of the IP
9	videos	Videos describing the IP and examples
10	vivado_package	Project folder for packaging the IP
11	vivado_project	Project folder for verification of the IP
12	xci	Folder containing xci files of IPs used in our design

Table 1: List of top level folders in delivered IP package

4 IP Package Content

Table 4 shows the list of folders in delivered IP package.

4.1 constraints

This folder contains constraints files used for pinning during design examples. Pinning constraint files are board depended. Our design examples are for the following boards:

1. Enclustra XU5 ZYNQ Ultrascale+
2. Digilent Arty A7
3. Digilent Nexys Video

4.2 doc

This folder contains the current user guide and also photos taken from hardware setups during tests or snapshots of vivado designs, simulation waveforms and similar items.

4.3 downloads

This folder contains datasheets and documents related to this IP. It is advised to always download the latest version directly from Internet.

4.4 packaged_ip

RGMII to AXI stream Vivado packaged IP is inside this folder. When the IP package is delivered, it contains this folder already populated with the packaged IP. You can include this folder in repositories of your Vivado project to use the IP in your Vivado block design.

If you made changes to the RTL source code of the IP (e.g. you added a new port) then you need to package the IP again. Refer to 4.11.1 for further description of packaging the IP in Vivado.

4.5 packaged_ip_rmii

Vivado packaged IP for talking to a RMII PHY is inside this folder. This folder is already populated when the IP package is dilevered. However, you can package the IP yourself too. Refer to 4.11.1 for further description of packaging the IP in Vivado.

4.6 tcl

This folder contains tcl scripts for packaging the IP as well as creating example vivado projects. The following list shows these scripts along with their description.

- `create_example_direct_sampling_zynquplus.tcl` : For XU5 board, Vivado design example shows how to use RGMII to AXI stream ip in direct sampling mode.
- `create_example_nexys_video.tcl` : For Digilent Nexys Video Artix 7 board, Vivado design example, IP is configured in RXC capture mode, link speed is 1 Gbits/s.
- `create_example_post_route.tcl` : Vivado project for post-route simulation of the IP
- `create_example_rxc_capture_100mbits_zynquplus.tcl` : For XU5 board, example Vivado project which uses the IP in 100 MBits/s mode.
- `create_example_rxc_capture_zynquplus.tcl` : For XU5 board, example vivado project which uses the IP in RXC capture mode. Link speed is 1 GBit/s.
- `create_example_with_dummy_modem_zynquplus.tcl` : Example vivado project which shows how the IP can be connected to other on-FPGA modules.
- `create_packaged_ip.tcl` : Packaging the IP can be done using this script.
- `create_packaged_ip_rmii.tcl` : Packaging the IP for RMII PHY can be done using this script.
- `create_example_rxc_capture_100mbits_rmii_artix7.tcl`: Creates an example design for Arty A7 board with RMII PHY.

4.7 verilog

IP source code (written in Verilog) is in this folder.

4.8 verilog_sim

Verilog simulation files of the IP are in this folder.

4.9 videos

Captured videos related to IP are in this folder.

4.10 vivado_project

Use this folder for creating Vivado example projects.

4.10.1 Howto Create Example Vivado Projects

In Windows run Vivado. After that, within Vivado, using the tcl console go to `vivado_project` folder. From there, source the tcl script that you want.

For example, imagine we have extracted the delivered file into:

```
C:\hdl\rgmii-to-axi-stream
```

We run Vivado and in tcl console we issue:

```
cd C:/hdl/rgmii-to-axi-stream/vivado_project
```

And then we issue:

```
source ../tcl/filename.tcl
```

Replace `filename.tcl` with the name of any tcl script you want to run.

In Linux, one usually sources Vivado's settings script and then changes directory into `vivado_project` and then runs Vivado using the following command:

```
vivado -source ../tcl/filename.tcl
```

4.11 vivado_package

Use this folder for packaging the IP. Refer to section 4.11.1 for further instructions.

4.11.1 Creating Packaged IP

The package is dilevered with RGMII to AXI stream IP already packaged inside `packaged_ip` folder. However you can re-package the IP as well. Use the `vivado_package` folder for packaging the IP. In Linux, go within that folder and run Vivado:

```
vivado -source ../tcl/create_packaged_ip.tcl
```

In windows run Vivado and then go to `vivado_package` folder (as described in section 4.10.1) and source the `create_packaged_ip.tcl` script.

If you are packaging the IP for talking to a RMII PHY, then do:

```
vivado -source ../tcl/create_packaged_ip_rmii.tcl
```

In this case the packaged IP will be created inside `packaged_ip_rmii` folder.

4.12 `xci`

XCI files are xilinx IP definition files. XCI files for different IPs used within the RGMII to AXI Stream bridge are in this folder.

5 Modes of Operation

The IP has 2 different operation modes:

1. Direct sampling
2. RXC capture

5.1 Direct sampling mode

In this mode the IP samples RX clock, RX valid and RX data lines all together using the provided `aclk_375MHz` pulse. That means the IP does not use RXC for capturing the data, instead it treats RXC just like a data line. The IP then internally selects proper received data bits based on edges of the captured RXC.

5.2 RXC capture mode

In this mode the IP uses RXC as a clock. This clock will be used for flip flops who capture the RX data and control lines.

5.3 Which Mode to Use?

For receiving data from PHY it is always better to use RXC capture mode when possible. However sometimes the pinning of FPGA is so that the RXC pin can not be used as a clock source in the design. In these situations direct sampling mode allows the designers to still operate the RGMII link.

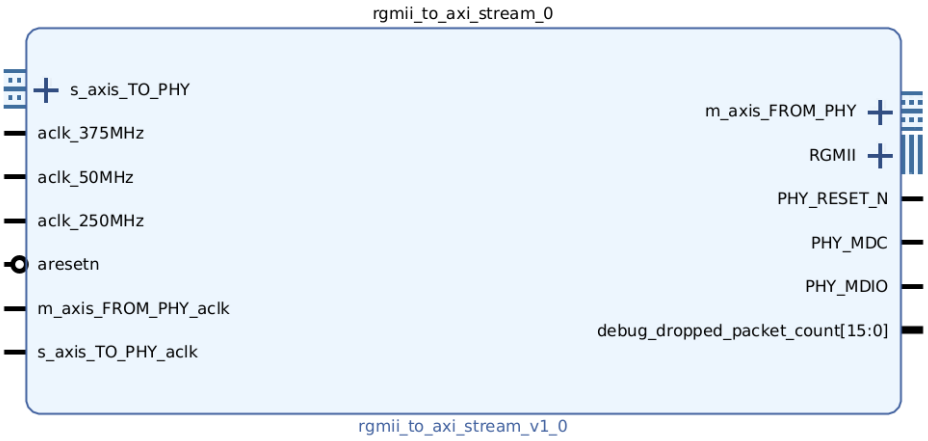

Figure 3: Symbolic representation of the IP in Vivado design flow. (RGMII PHY)

index	Interface Name	Direction	Description
1	m_axis_from_PHY	Master	Received ethernet packets from PHY
2	s_axis_TO_PHY	Slave	Ethernet packets to PHY
3	RGMII	Master	RGMII signals to and from the PHY
4	RMII	Master	RMII signals to and from the PHY

Table 2: List of interfaces of the IP

6 Ports and Interfaces

Figure 3 shows a symbolic representation of the IP in Vivado block design flow when connecting to a RGMII PHY. Figure 4 shows a symbolic representation of the IP in Vivado block design flow when connecting to a RMII PHY. As can be seen the IP has fewer clock ports when connecting to a RMII PHY.

6.1 Interfaces

Table 6.1 describes the interfaces of the IP. Both of `m_axis_from_PHY` and `s_axis_TO_PHY` use AXI stream to transfer packets between the IP and other modules. RGMII interface is used when the IP is packaged for talking to a RGMII PHY. RMII interface is used when the IP is packaged for talking to a RMII PHY.

6.1.1 `m_axis_from_PHY`

Ethernet packets received by PHY from the network will be passed to our IP using RGMII interface. Our IP captures these packets, checks their CRC and passes these packets to next FPGA module using this AXI stream interface.

Figure 4: Symbolic representation of the IP in Vivado design flow. (RMII PHY)

Notes about this interface:

1. The data (`tdata`) width of this AXI stream interface is by default 1 bytes.
2. `TUSER` signal indicates the CRC status of the received packet. If it is high while the IP is sending the packet out, that means packet has a CRC error.
3. This interface operates at `m_axis_from_PHY_aclk` clock domain.

6.1.2 `s_axis_TO_PHY`

Ethernet packets which should be sent over network will be given to our IP by other on-FPGA modules, through this interface. Our IP receives the ethernet packet and then sends it to PHY using the RGMII interface.

Notes about this interface:

1. The data (`tdata`) width of this AXI stream interface is by default 1 bytes.
2. This interface operates at `m_axis_TO_PHY_aclk` clock domain.

6.1.3 CRC calculation for `m_axis_FROM_PHY`

The IP calculates the CRC of each of the arrived packets from the PHY. As described `TUSER` is used to indicate if a packet has CRC error.

6.1.4 CRC calculation for `m_axis_TO_PHY`

The IP can calculate the CRC for packets before sending them to PHY. The IP also allows the user to do it himself.

In Figure 7 there is a configuration option `Tx Add CRC to Packet`. When set to `TRUE` the IP will receive your ethernet packet, will calculate its CRC and will send the result to the PHY. When set to `FALSE` the IP assumes the arrived ethernet packet from the previous module already contains the calculated CRC.

index	Signal Name	Direction	Description
1	RXD[3:0]	Input	RX Data
2	RXC	Input	RX Clock
3	RXCTL	Input	RX Valid and Control
4	TXD[3:0]	Output	TX Data
5	TXC	Output	TX Clock
6	TXCTL	Output	TX Valid and Control

Table 3: List of signals used in RGMII interface.

index	Signal Name	Direction	Description
1	RXD[3:0]	Input	RX Data
2	RXC	Input	RX Clock
3	RXCTL	Input	RX Valid and Control
4	RXER	Input	RX error
5	TXD[3:0]	Output	TX Data
6	TXC	Input	TX Clock
7	TXCTL	Output	TX Valid and Control

Table 4: List of signals used in RMII interface.

6.1.5 `RGMII`

Table 3 lists the RGMII signals between our IP and any RGMII PHY.

6.1.6 `RMII`

Table 4 lists the RMII signals between our IP and any RMII PHY.

6.2 Ports

Table 5 describes the rest of the ports of the IP.

6.3 Clocks and Resets

Table 6 shows the clock ports of the IP. It also shows the reset.

index	Port Name	Direction	Description
1	PHY_MDC	IO	MDIO Clock port
2	PHY_MDIO	IO	MDIO Data port

Table 5: List of ports of the IP

index	Port Name	Description
1	aclk_50MHz	50 MHz clock, used for MDIO
2	aclk_250MHz	TX clock used to send packets to PHY. see section 6.3.1
3	aclk_375MHz	RX clock used to receive packets from PHY. see section 6.3.2
4	aresetn	active low reset, resets the entire IP.

Table 6: List of clocks and resets of the IP

6.3.1 TX Clock

The name chosen for TX clock port is `aclk_250MHz` however, in practice this port does not necessarily need a 250 MHz clock. Here are possible scenarios:

- At **1 Gbits/s** the clock should be 250 MHz.
- At **100 MBits/s** the clock should be 50 MHz.

When the IP is using RMII interface this clock port does not exist.

6.3.2 RX Clock

The name chosen for RX clock port is `aclk_375MHz` however, in practice this port does not necessarily need a 375 MHz clock. Here are possible scenarios:

- In **direct sampling mode at 1 GBits/s**5.1 the clock should be 375 MHz.
- In **RXC capture mode at 1 GBits/s** 5.2 the clock should be 125 MHz or any bigger number. It is recommended to use a higher frequency than 125 MHx.
- At **100 MBits/s** the clock should be 50 MHz or bigger.

When the IP is using RMII interface this clock port does not exist.

6.3.3 AXI Streams Clocks

The IP has 2 AXI streams, one for recevinging packets from other FPGA modules and the another for sending packets to these modules. Each of these AXI interfaces can operate in a different clock domain. The names of IP's clock ports are `m_axis_FROM_PHY_aclk` and `s_axis_TO_PHY_aclk`.

7 IP Parameters and Configuration (RGMII PHY)

Figures 7 shows the GUI of the IP.

In the following we go through each of the configuration options.

7.1 PHY Type

Currently the following list of PHYs are supported and tested.

- Microchip KSZ9031RNX tested on Enclutra XU5 board.
- RealTek RTL8211E tested on Kintex Ultrascale board.
- Microchip VSC8541XMV tested on Digilent Nexys Video board.

Important facts is all of the above PHYs have a similar register address map. Basically, any other PHY with standard MDIO address map is supported too.

7.2 PHY Address

Important parameter specifying the address of the PHY.

7.3 Force 100MBits Link

if `TRUE` the IP will program the PHY after reset and forces it to operate at 100 MBits/s. Set this option to `FALSE` when you want to have your link running at 1 GBits/s.

7.4 Loopback Enable

If set to `TRUE` the IP will program the PHY and enables digital loopback mode.

7.5 Use RX clock to capture data

When `TRUE` the IP will use `RXC` input as the clock to capture the incoming data and valid lines. When `FALSE` the IP will be in direct sampling mode where it treats all of the signals equally. It samples all of them and then by looking at positive and negative edges of `RXC` it determines the correct value of data bits. This mode is good if you have limitations with FPGA pin assignment and `RXC` is not located on a pin which can be a used a local or global clock.

7.6 Set RX skew

If `TRUE`, after reset the IP will configure the PHY to delay each of the RX control and data lines with regard to RX clock so that they can be safely captured at FPGA side. This is suitable when the IOs you are using at FPGA side for receiving signals from the PHY are not in the High Performance (HP) IO banks and so they do not have the internal IDELAY elements.

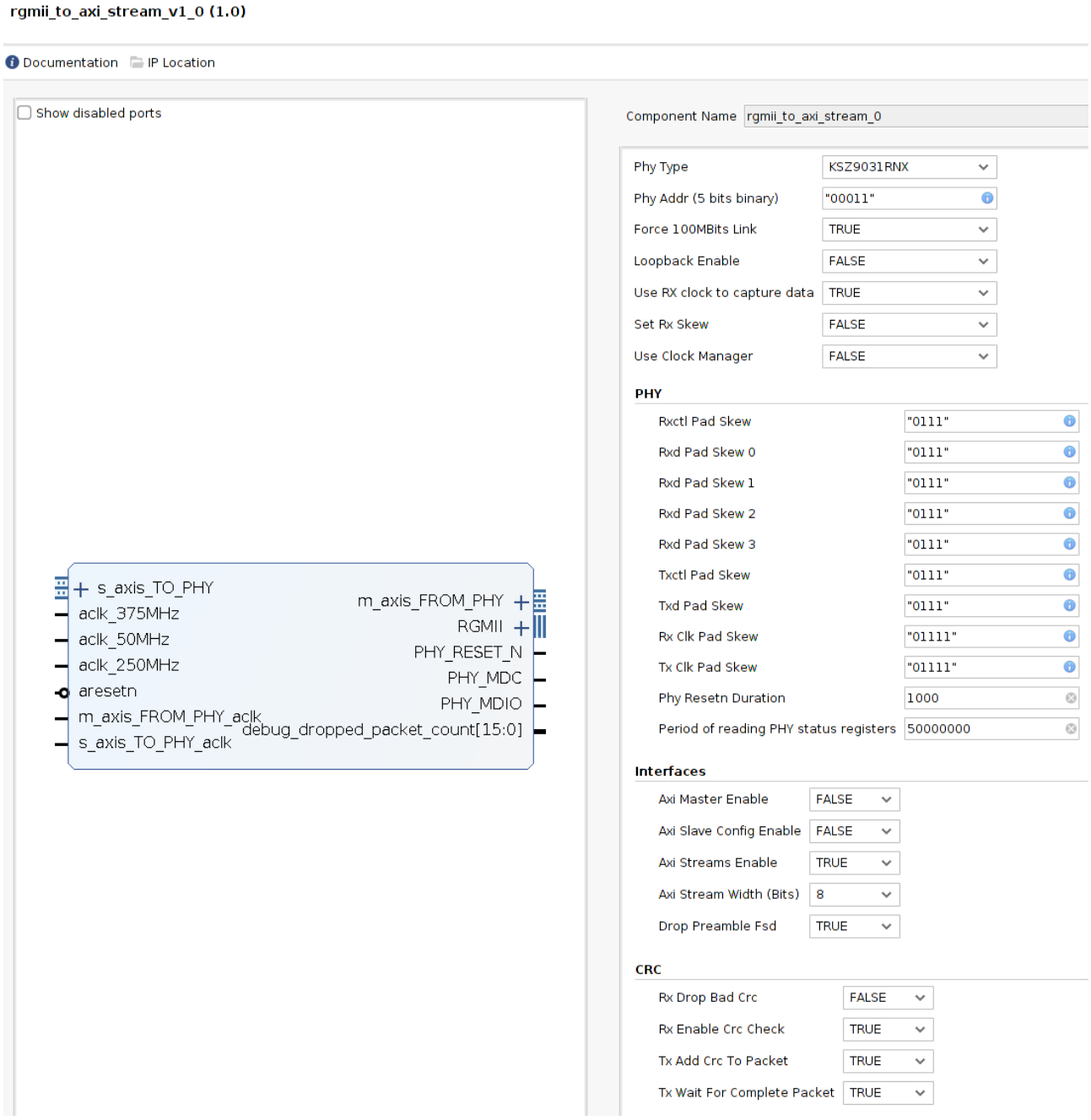

Figure 5: IP parameters configuration GUI. In this setup, IP is configured to force the PHY to 100MBits ethernet.

7.7 Use Clock Manager

If `TRUE` the IP will instantiate a clock management tile on RX clock path. The IP will then receive the RX clock from PHY and first it passes it through a clock management tile where it creates a clock with equal phase to RX clock. The generated clock by clocking wizard will then be used for capturing the data.

7.8 Skew Values

User can set the amount of skew which should be programmed into PHY for each of the following items. The value should be a binary.

- RX Control
- RX Data
- TX Control
- TX Data
- RX Clock
- TX Clock

Note that for RX Data lines, user can set the skew of each line separately and they may not be equal. For TX Data lines one skew will be set for all line.s

7.9 PHY Reset Duration

Specifies how long should the IP keep the PHY in reset (in MDIO clock cycles).

7.10 Period of Read PHY Status

The IP can read and report PHY registers continously. This value indicates how often should IP read and report register values. Reporting values can be through VIO or AXI slave registers.

7.11 AXI Master Enable

In advanced version of the IP, it can directly put received Ethernet packets into destination DRAM memory or read Ethernet packets form DRAM memory and send them to the PHY. These data transfer will be done using IP's internal DMA engine.

7.12 AXI Slave Config Enable

Enables the AXI slave interface and allows the user to read IP registers through this interface and also perform MDIO read and write accesses.

7.13 AXI Streams Enable

Enables AXI stream interfaces of the IP. These are the main interfaces through which the IP receives and sends Ethernet packets. Sections 6.1.1 and 6.1.2 describe these interfaces in more detail.

7.14 AXI Stream Width (Bits)

Width of AXI stream `TDATA` interface. Typical value is 1 Bytes. It can be 2, 4 or 8 bytes as well.

7.15 Drop Preamble FSD

Typical behavior of the IP is to drop the preamble and FSD of the Ethernet packets it receives.

7.16 RX Drop Bad CRC

When `TRUE` IP drops the recieved ethernet packets which have incorrect CRC.

7.17 RX Enable CRC Check

When `TRUE`, enables the CRC calculation engine of the IP.

7.18 TX Add CRC to Packet

When `TRUE` for each ethernet packet that IP should send to PHY, the IP calculates the CRC and adds the CRC to tail of the packet.

7.19 TX Wait for Complete Packet

When `TRUE` the IP will wait until the complete TX packet arrives (from the upstream FPGA module) and then starts sending the packet over Ethernet interface. This is specially helpful if the FPGA module who produces the packets to be sent, does not create conitnous packets and data flow interrupts time to time.

7.20 IP Parameters and Configuration (RMII PHY)

Figures 6 shows the GUI of the IP when connecting to RMII PHY.

Description of the parameters is similar to RGMII mode.

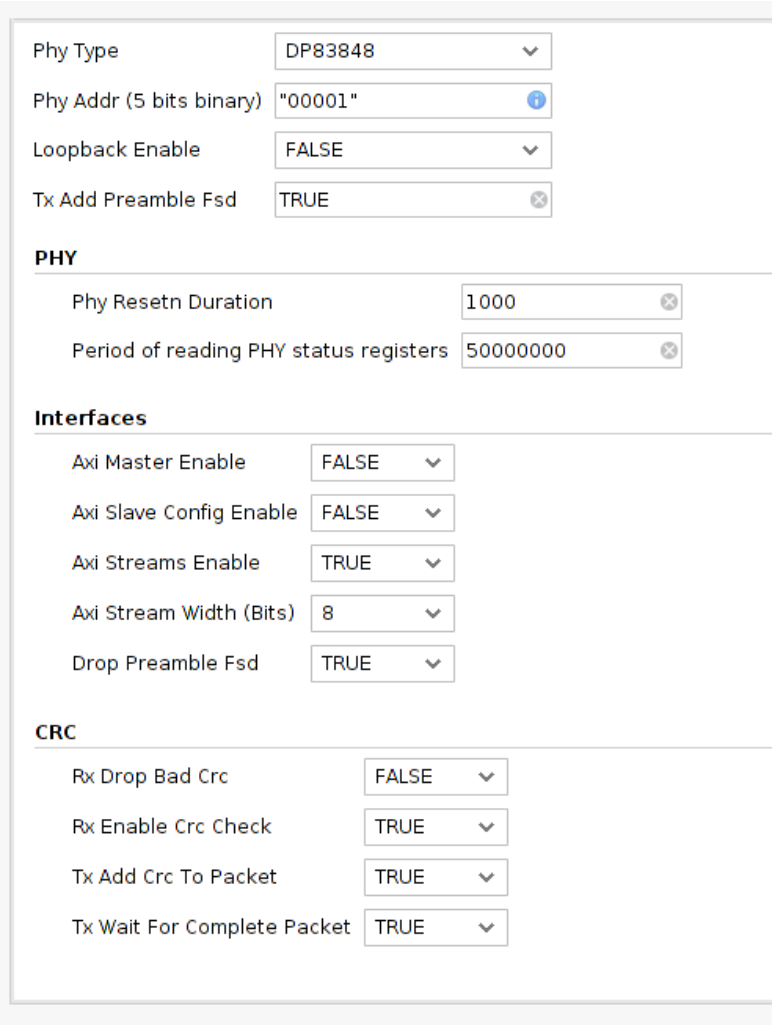

Figure 6: IP parameters configuration GUI. In this setup, IP is configured to connect to a RMII PHY.

8 Registers

By default the AXI slave lite interface of the IP is disabled. User does not need any specific register access through AXI Lite interface to confiugre the IP.

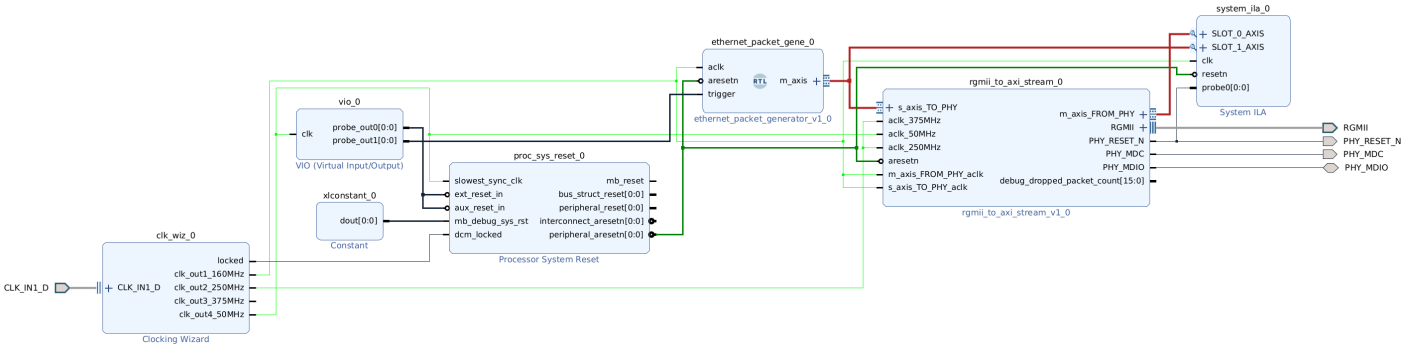

Figure 7: Vivado block design of RXC capture design example. 1 GBits/s link speed.

9 Exmaple Designs

The package comes with various example designs. In Vivado block designs that we show for each of the examples, we have the following color coding:

- Clock signals are in light green
- Reset in dark green
- interface connections in red.

Note that the same design example is available for multiple boards.

9.1 RXC capture

Figure 9.1 shows the block design for this example.

You can create this design example with each of these two tcl scripts:

- `create_example_post_route_rxc_capture_1GBits.tcl` creates the example design for link speed of 1 GBits/s.
- `create_example_post_route_rxc_capture_100MBits.tcl` creates the example design for link speed of 100 MBits/s.
- `create_example_rxc_capture_100mbits_rmii_artix7.tcl` creates example design to connect to RMII PHY on Arty A7 Artix board.

Figure 9.1 shows the block design. RGMII to AXI stream IP is instantiated and is connected to outside world through RGMII interface and MDIO signals. An Ethernet packet generator is also instantiated. It produces Ethernet packets to be sent over the IP to the network. Ethernet packet generator can produce example ARP packets. Section 9.3 provides more details. A VIO controls the reset of the system and also start of packet generation. A system ILA monitors the AXI stream interface of the IP.

In Figure 9.1 the IP is set to operate at 1 GBits/s. Clocks for AXI interfaces are being driven by a 160 MHz clock source. It is recommended that the clock for AXI interfaces be higher than 125 MHz. A 50 MHz clock is used for MDIO connectivity. A 250 MHz clock is used for both TX and RX paths inside the IP. As we are using the IP in RXC capture mode, we do not need a 375 MHz clock.

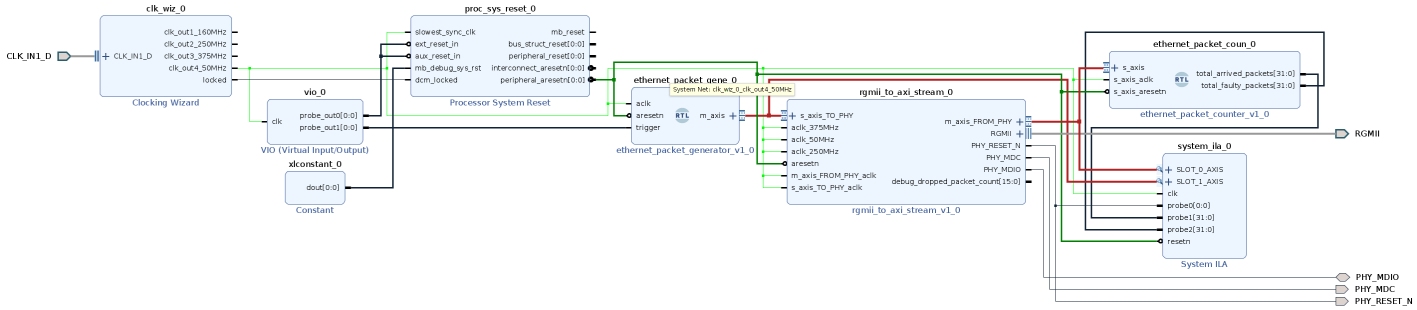

Figure 8: Vivado block design of RXC capture design example. 100 MBits/s link speed.

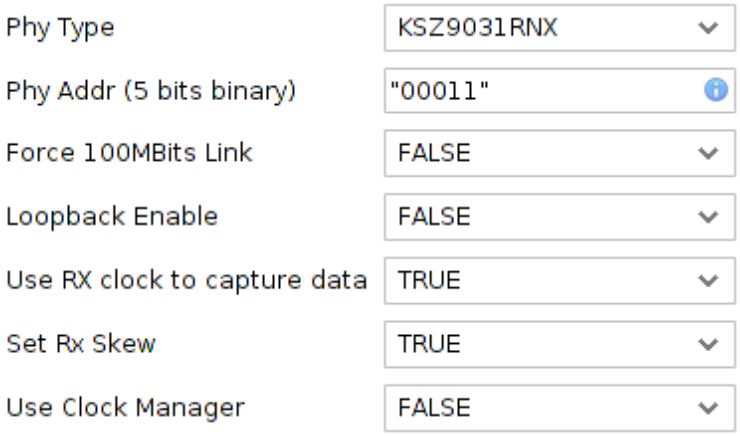

Figure 9: IP configuration 1 GBits/s link.

Figure 8 shows a similar example. This time the IP is in 100 MBits mode. The entire design uses a 50 MHz clock.

Figure 9 shows IP configuration in this design example when 1 GBits mode is active. As can be seen, we ask the IP to program the PHY and configure the skew registers. Note that enabling `Set RX Skew` option makes the PHY to program the skew registers for both of RX and TX. Section 9.4 provides further description of skew registers.

Figure 10 shows the same configs when IP should operate at 100 MBits.

Figure 11 shows VIO setup for our example. There exist two VIOs in the design. One is responsible for controlling the reset (probe0) and generation of packets (probe1).

Figure 12 shows captured ILA waveforms when a new packet has arrived. As can be seen, ILA captures the signals of both incoming and outgoing AXI streams. It also captures total number of packets arrived and total number of faulty packets arrived.

Figure 13 shows captured ILA waveforms in RMII PHY design example when a packet is being trasmitted to network.

Figure 14 shows a snapshot of Wireshark software running on the PC. As can be seen whireshark has captured the packets that our packet generator IP on the FPGA produced.

9.2 Enabling Loopback Mode

Remember, you can always put the PHY into Digital loopback mode easily. All you need to do is the put the config value for loopback to True in IP configuration GUI. This is usually useful

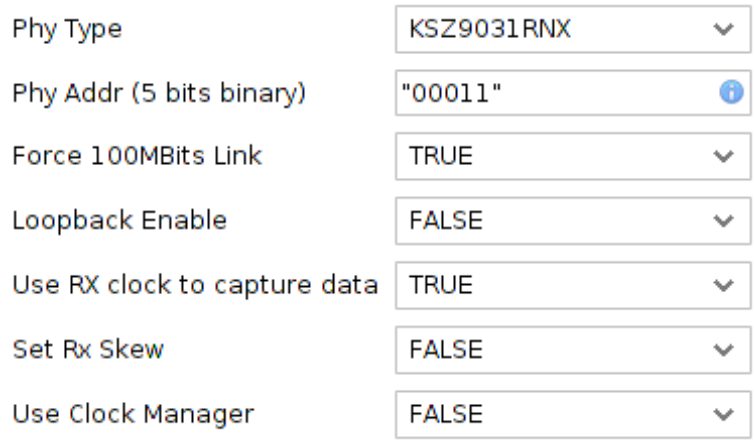

Figure 10: IP configuration 100 MBits/s link.

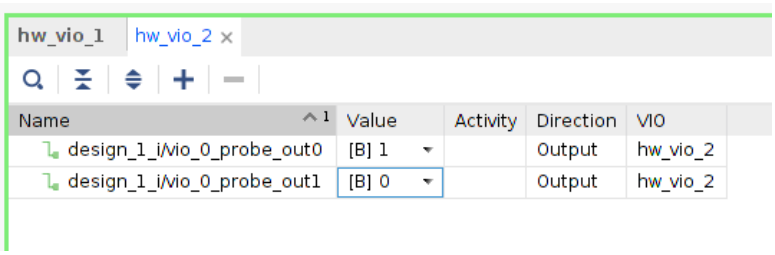

Figure 11: VIO setup for controlling reset and generation of packets.

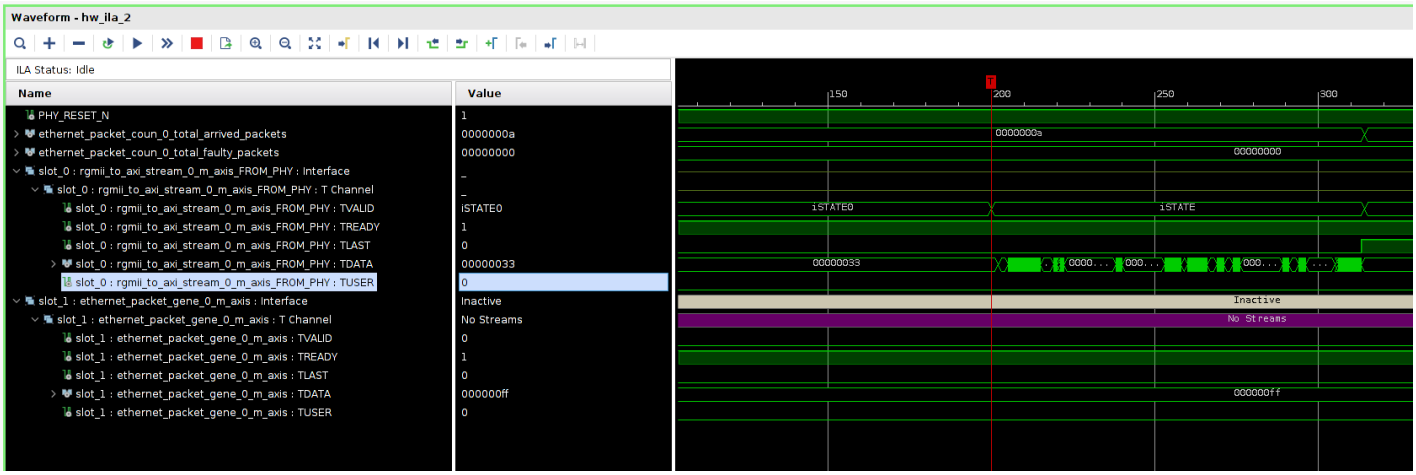

Figure 12: Captured ILA waveform of an arriving packet.

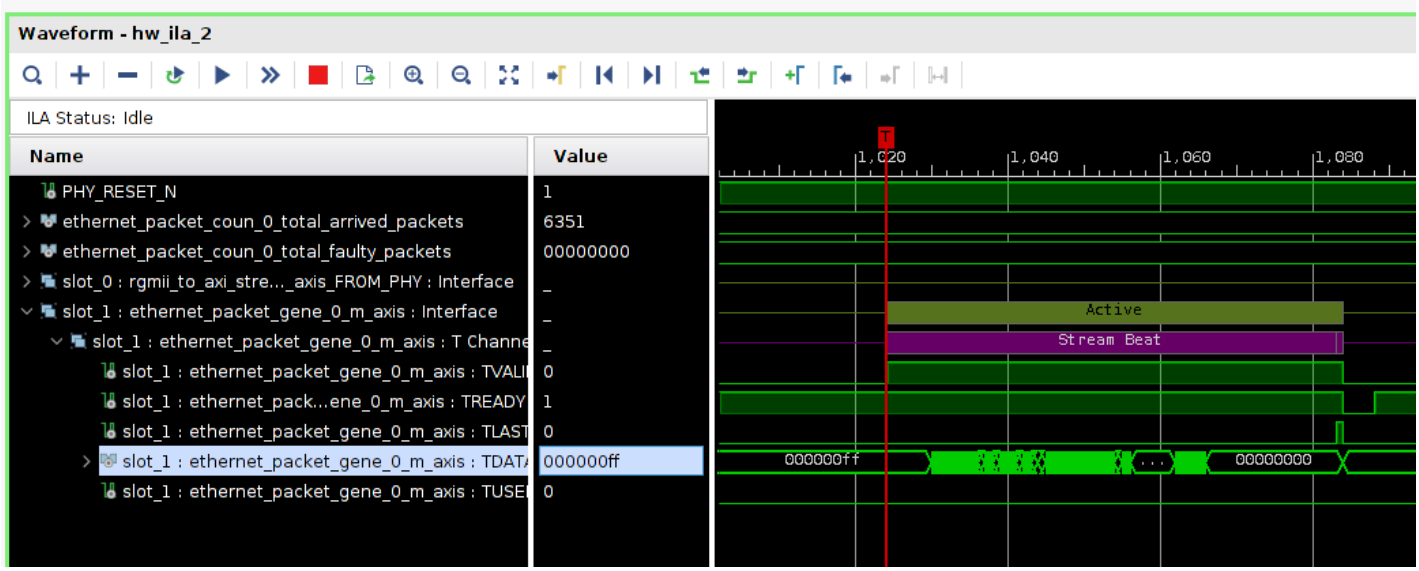

Figure 13: Captured ILA waveform of transmitting a packet (RMII mode).

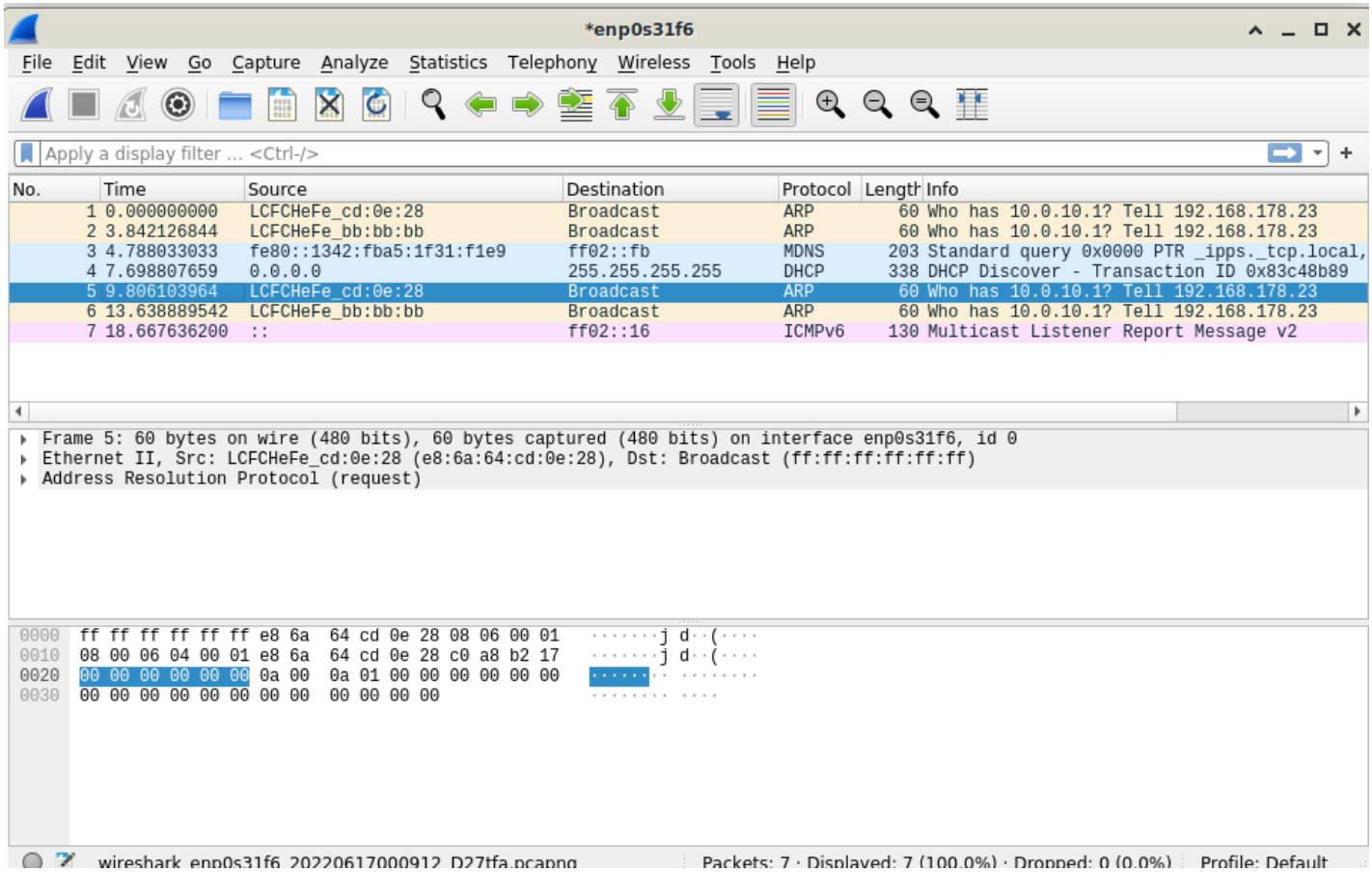

Figure 14: Wireshark capturing ARP packets produced by our packet generator IP. (Arty A7 board, RMII PHY)

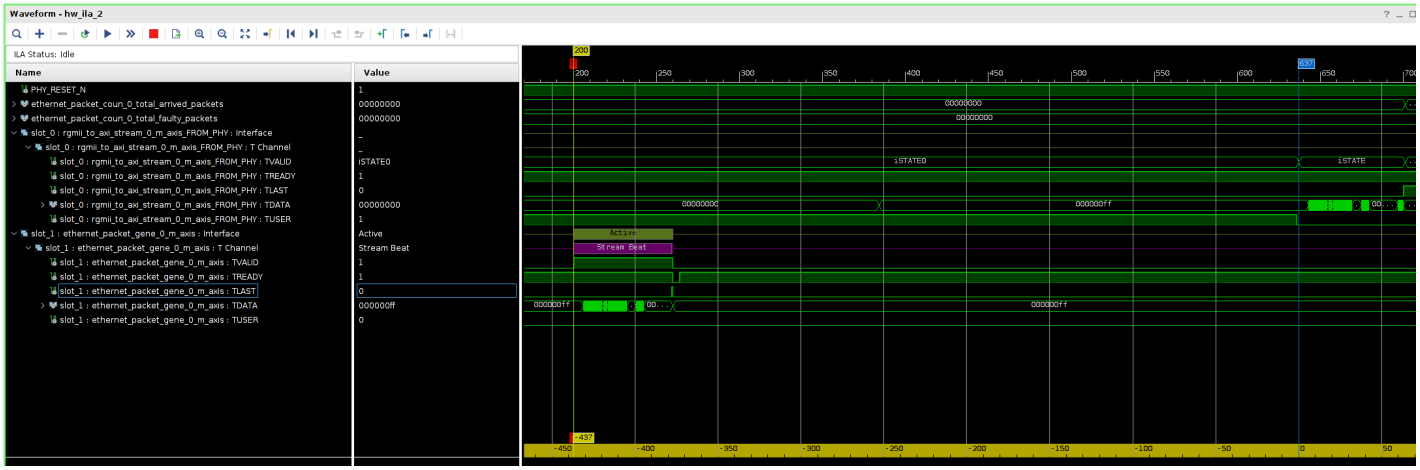

Figure 15: Example ILA waveform capture, loopback is enabled.

when you have doubts about the connection between FPGA and PHY or its timing.

By enabling the loopback mode, and configuring system ILA to start capturing the data when Trigger signal to packet generator goes high, you will be able to see both the packet which goes from packet generator to RGMII to AXIS IP and then the packet that gets reflected back to FPGA by the PHY.

The two packets should be identical. Only the CRC should be added. Remember RGMII to AXIS adds the CRC of the packet to its tail itself.

An on board waveform capture using ILA when loopback mode is enabled is shown in Figure 9.2. As can be seen a packet starts getting trasmitted at clock cycle 200. At clock cycle 637, the looped back packet comes back. This is a total latency of around 850 micro-seconds for digital loopback for the RGMII to AXIS IP and the PHY together.

9.3 Packet Generator

Packet generator is a Verilog module located inside `verilog_sim` folder. It is fully synthesizable and can be used in Vivado designs. It contains a memory which holds the packets that user has defined.

Inside `verilog_sim` folder there is a memory initialization file `01.mem`. This file can be editted and any desired packet can be added.

Packet generator then starts putting out packets with any positive or negative edge of `trigger` signal.

9.4 Programming Skew Registers

In 1 GBits/s mode, when enabled in the GUI, the IP can proram PHY's RX and TX skew registers itself right after reset. The amount of skew in RX data lines and clock can be defined separately for each line in the IP GUI. For TX, a same skew value will be applied to all TX data lines and a separate value will be applied to clock skew.

10 Tested Boards

This IP is tested on following boards:

index	Board	Device	PHY
1	Enclustra XU5	Zynq Ultrascale+	Microchip KSZ9031RNX
2	Digilent Nexus Video	Artix7	RealTek RTL8211E-VL
3	Digilent Arty A7	Artix7	Texas Instruments

11 Revisions

1. Initial version created on 30th of April, 2021.
2. Update to IP documentation on 26th of May, 2021. Ports and interfaces, example designs added.
3. Update to IP docuemntation, added more examples, ila waveforms, block designs. June 12th, 2022.
4. IP now supports PHYs with RMII interface for 100MBits Ethernet. June 17th, 2022.