XAPP1305 (v1.4) April 30, 2019

PS and PL-Based 1G/10G Ethernet Solution

Authors: Bhargav Shah, Naveen Kumar Gaddipati, Akhilesh Mahajan, and Rhythm Jain

Summary

This application note focuses on Ethernet-based designs that use Zynq® UltraScale+™ devices. It describes the use of the gigabit Ethernet controller (GEM) available in the processing system (PS) through the extended multiplexed I/O (EMIO) and multiplexed I/O (MIO) interfaces. It also describes the use of 1000BASE-X, SGMII, and 10GBASE-R physical interfaces using high-speed transceivers in programmable logic (PL). The use of Ethernet jumbo frames in both PS and PL-based Ethernet systems is explained in this application note. Throughput numbers for PS Ethernet, PL Ethernet (1G and 10G), and PS-PL Ethernet are also included. The designs explained in this application note demonstrate Ethernet solutions with kernel-mode Linux device drivers.

Download the reference design files for this application note from the Xilinx website. For detailed information about the design files, see Reference Design.

Introduction

Zynq UltraScale+ devices integrate a flagship ARM® Cortex®-A53 64-bit quad-core or dual-core processor, Cortex-R5 dual-core real-time processor in PS, and PL in a single device.

The PL includes the programmable logic, configuration logic, and associated embedded functions. The PS comprises the ARM Cortex-A53 MPCore CPUs unit, Cortex-R5 processors, on-chip memory, external memory interfaces, cache coherent interconnect (CCI), and peripheral connectivity interfaces. The PS is equipped with four GEMs. Each controller can be configured independently and uses a reduced gigabit media independent interface (RGMII). The RGMII interface is routed through MIO pins to interface with an external RGMII PHY. Other Ethernet communications interfaces such as TBI, RGMII v2.0, and SGMII can be created in the PL using the GMII/MII available on the EMIO interface.

SGMII is also supported by the GEM using the PS-GTR transceiver without using any logic in the PL. Figure 1 shows the various Ethernet implementations on the ZCU102 board.

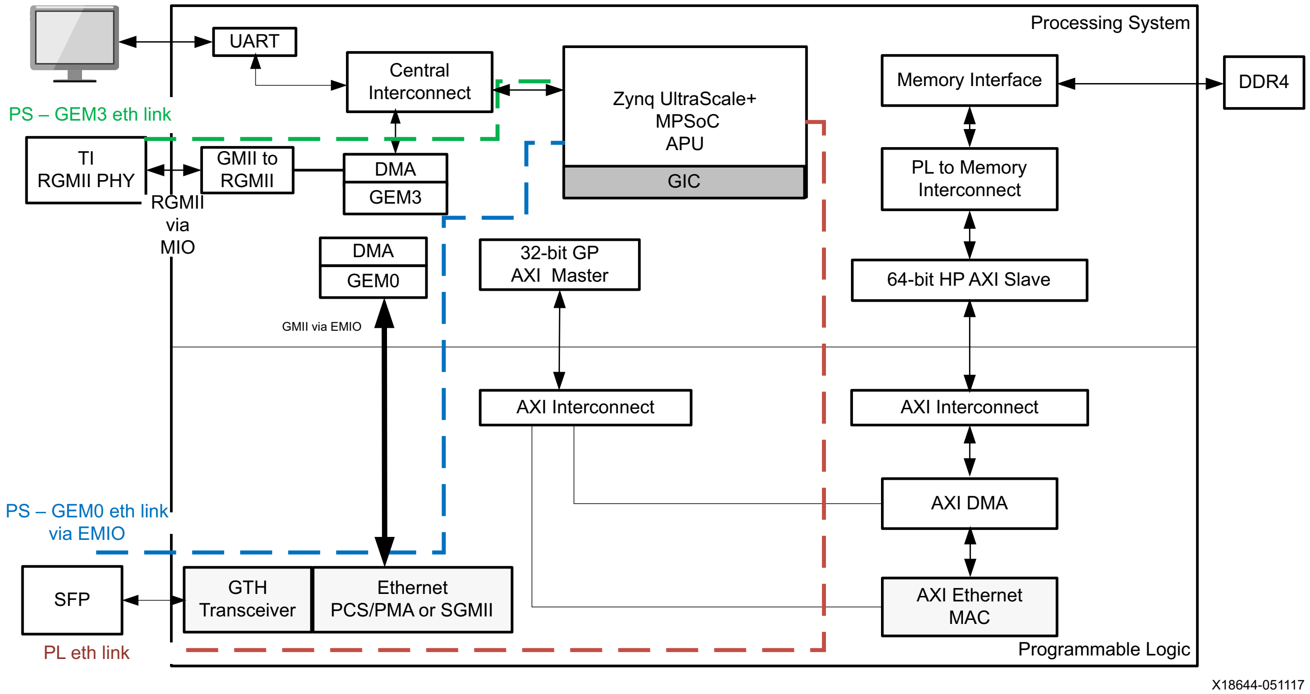

Figure 1: **Zynq UltraScale+ MPSoC Ethernet Interface**

Note: The PS-GEM3 is always tied to the TI RGMII PHY on the ZCU102 evaluation board. The 1000BASE-X/SGMII PHY and the GTH transceiver are a part of the AXI Ethernet core for 1G PL Ethernet link, which uses the AXI 1G/2.5G Ethernet subsystem IP core [Ref 1]. The PS-PL Ethernet uses PS-GEM0 and 1G/2.5G Ethernet PCS/PMA, or SGMII core [Ref 2]. The 10G PL Ethernet link uses 10/25G high-speed Ethernet subsystem IP core [Ref 3].

In the designs provided with this application note, the PS-GEM3 is connected to the Texas Instruments DP83867IRPAP Ethernet RGMII PHY device through the reduced gigabit media independent interface (RGMII). This is the default setup for the ZCU102 board. This application note demonstrates various PS and PL-based Ethernet implementations. The designs described in this application note are listed below.

- PS Ethernet (GEM3) connected to a 1G physical interface in PS through an MIO interface. See Using PS GEM through MIO.
- PS Ethernet (GEM0) connected to a 1000BASE-X/SGMII physical interface in PL through an EMIO interface. See Using PS GEM through EMIO.
- Ethernet implemented as soft logic in PL (MAC) and connected to the 1000BASE-X/SGMII physical interface in PL. See Using PL 1G Ethernet.
- Ethernet implemented as soft logic in PL (MAC) and connected to the 10G physical interface in PL. See Using PL 10G Ethernet.

Note: GEM0, GEM1, or GEM2 can also be used for PS Ethernet. The hardware design varies depending on the GEM selected.

Using PS GEM through MIO

This section describes how to use the PS Ethernet block GEM3 with the PS PHY through the MIO interface.

Hardware Design

The PS Ethernet controller (GEM3) connects the on-board TI PHY through MIO pins using the RGMII interface. The GEM3 block is enabled while generating the hardware system. The GEM3-TI PHY link is shown in Figure 1 with the PS-GEM3 link. For more information refer to the *PS and PL based Ethernet in Zynq MPSoC wiki* [Ref 4].

Reference Clock Generation

The Ethernet reference clock (125 MHz) for each of the GEMs is generated by configuring the internal PLL of the PS.

Software Design

This design uses the common **`macb.c`** driver code (present in the Linux kernel) for all the GEMs on the ZCU102. The macb driver uses the direct memory access (DMA) controller attached to the GEM in the PS. This driver is responsible for several functions including DMA descriptor rings setup, allocation, and recycling. The interrupt handling is done only for the PS GEM events because the interrupt status implicitly reflects DMA events. Additionally, the device tree is updated to include PS-GEM3 with relevant parameters. Refer to Device Trees for more information.

Linux Driver

A monolithic Linux device driver is used in this design. The software architecture for PS Ethernet interfaces is shown in Figure 2.

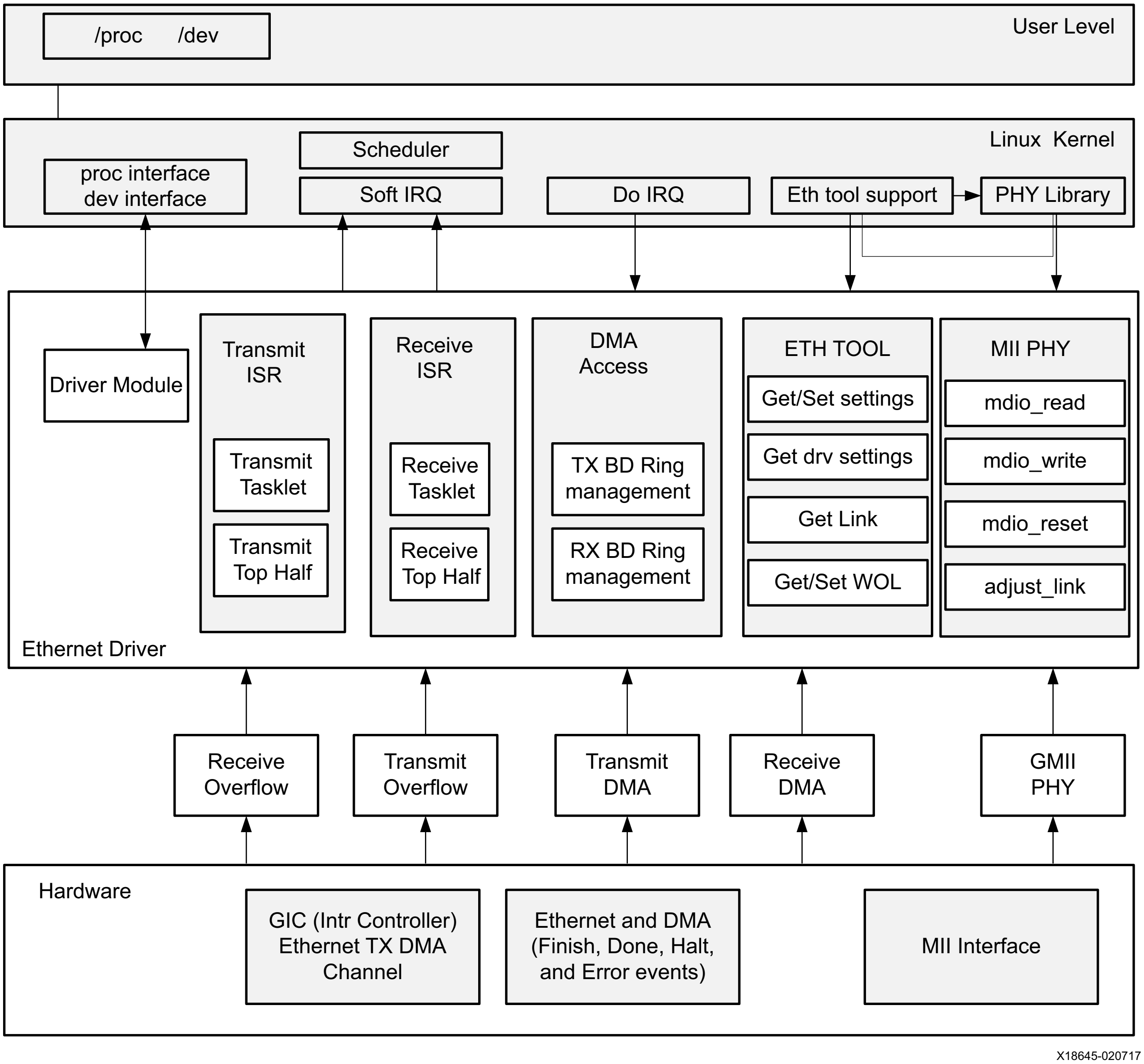

Figure 2: **PS Ethernet Software Driver for Linux**

Using PS GEM through EMIO

This section describes the use of the PS Ethernet block GEM0 with the PL PHY through the EMIO interface. The PS GEM block can be accessed through the PL using EMIO pins that allow GMII and management data input/output (MDIO) interfaces to be connected to the physical layer. The 1G/2.5G Ethernet PCS/PMA or SGMII core can be used as the physical media for the Ethernet in 1000BASE-X or SGMII mode. High-speed serial transceivers are used to access the small form factor pluggable (SFP) cage on the ZCU102 board. The SFP cage is connected to a standard Ethernet LAN through an SFP-to-RJ45 converter module. To enable the SFP, jumper J16 should be shorted as shown in Figure 7.

Hardware Design

As shown in Figure 3, the GMII interface connects the PHY and PS GEM through the EMIO pins. The GEM0 block is enabled while generating the hardware system in the Vivado® tools. The PHY address port of 1G/2.5G Ethernet PCS/PMA or serial gigabit media independent interface (SGMII) core can be assigned a fixed value in the range of 1 to 31. See the *PS and PL based Ethernet in Zynq MPSoC wiki* [Ref 4] and *1G/2.5G Ethernet PCS/PMA or SGMII v16.0 LogiCORE IP Product Guide* (PG047) [Ref 2] for more information.

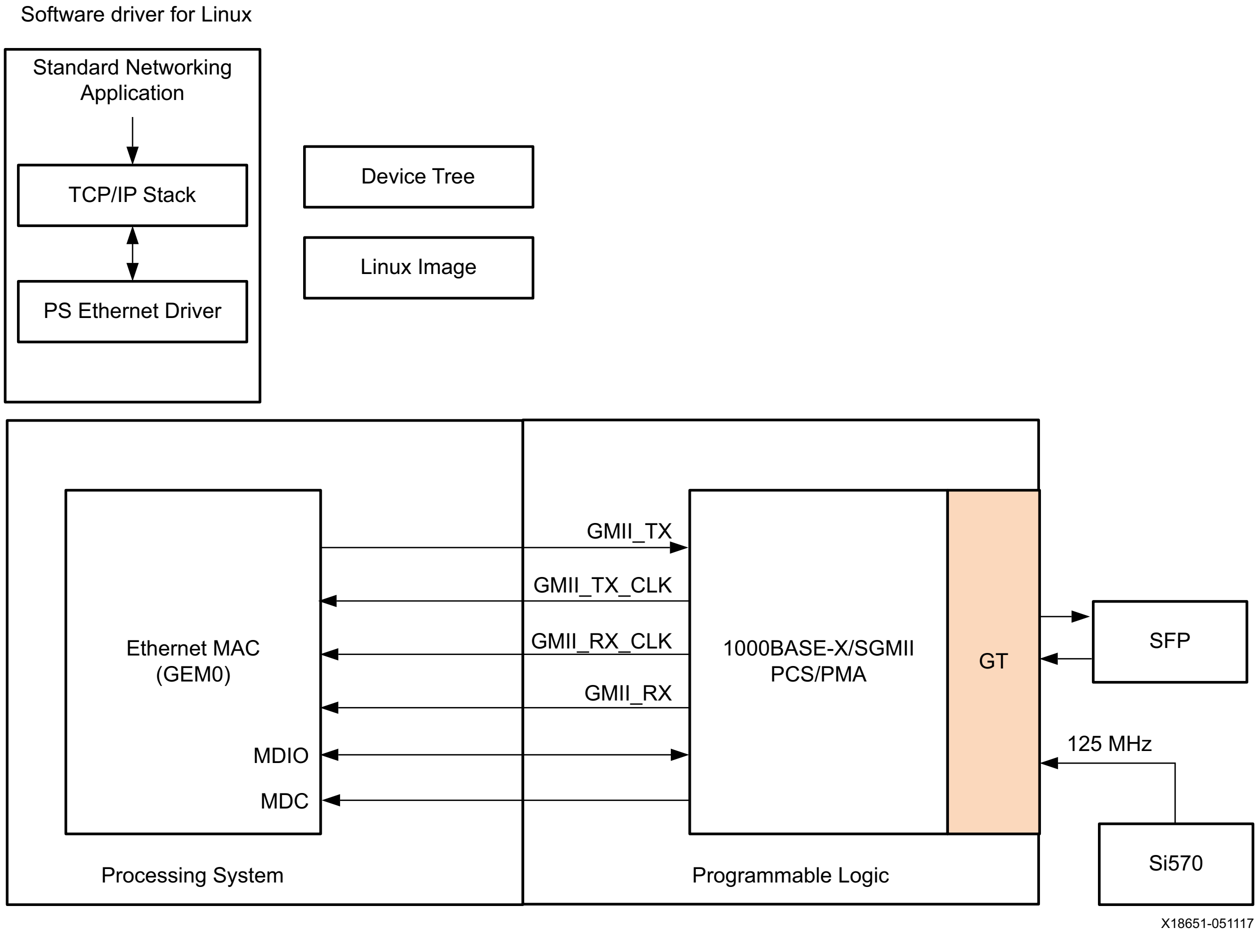

Figure 3: **PS-PL Ethernet Design**

Reference Clock Generation

The GTH transceiver X0Y4 on the Zynq UltraScale+ MPSoC is connected to the SFP cage on the ZCU102 board. The GTH transceiver reference clock (125 MHz differential) is generated from the Si570 jitter attenuator on the ZCU102 board. The clock divider values are adjusted to generate 125 MHz from the Si570 programmable oscillator. The Si570 is programmed over the I2C interface to generate the required clock value. See the Si 570 data sheet [Ref 5] for details on the Si570.

To enable GEM0 through the EMIO interface, specific registers must be programmed. This is part of the PS configuration data used by the Zynq UltraScale+ MPSoC first stage boot loader (FSBL).

To select the EMIO as the source for receiving clock, data, and control signals, set the SLCR.GEM0_CLK_CTRL[SRCSEL] bit to **`3'b1xx`**, where **`x`** is a don't care (1 or 0).

Software Design

This design uses the common **`macb.c`** driver code for all the GEMs on the ZCU102. The macb driver uses the DMA controller attached to the GEM in the PS. This driver is responsible for several functions, including DMA descriptor rings setup, allocation, and recycling. The interrupt handling is done only for the PS GEM events, because the interrupt status implicitly reflects DMA events. Additionally, the device tree is updated to include PS-GEM0 with relevant parameters. Refer to Device Trees for more information.

Note: To support other PL physical interfaces such as TBI, the hardware design and device tree must be edited. The PHY specific initialization is handled by the phylib subsystem in the Linux driver (macb), and information regarding the PHY can be provided in the device tree. To use the phylib subsystem for PHY programming, the phylib subsystem must support the PHY initialization routine for the desired PHY.

Linux Driver

A monolithic Linux device driver is provided for this design. Figure 2 shows the software architecture for the PS Ethernet interfaces.

Using PL 1G Ethernet

This section describes the PL implementation of the Ethernet. This design consists of the AXI 1G/2.5G Ethernet subsystem, AXI DMA, and AXI Interconnect IP cores. The AXI 1G/2.5G Ethernet subsystem IP core consists of tri-mode Ethernet MAC (TEMAC) and 1G/2.5G Ethernet PCS/PMA or serial gigabit media independent interface (SGMII) cores. A high performance (HP) port is used in this design for fast access to the PS-DDR memory. The general-purpose slave port can also be used if the HP port is occupied with other peripherals.

Hardware Design

Ethernet implementation in the PL is shown in Figure 4. The HP port is used for fast data transfers between the PL and the PS-DDR4 memory. It connects to the AXI DMA scatter-gather, stream to memory mapped (S2MM) and memory mapped to stream (MM2S) interfaces through the AXI interconnect. This interconnect also performs data width conversion to connect the 64-bit HP port to the 32-bit interfaces of the AXI DMA. In the AXI DMA, both the scatter-gather option and data realignment engine are enabled for the S2MM and MM2S paths.

The streaming interface of the AXI DMA is connected to the AXI Ethernet subsystem. The AXI Ethernet subsystem has full checksum offloading (CSO) enabled and has FIFO depths of 32K to support jumbo frame transfers.

The AXI Ethernet core implements an Ethernet MAC and supports 1000BASE-X and SGMII PHY interfaces. It connects to the SFP through GTH transceivers through 1000BASE-X/SGMII interfaces.

For the control interface, a general purpose (GP) AXI master port is enabled in the PS. This port connects to the AXI DMA and AXI Ethernet cores.

The 1000BASE-X and SGMII PHY registers are accessed using the MDIO interface provided by the AXI Ethernet core. The interrupt ports from the AXI DMA and the AXI Ethernet IP cores are connected to the general interrupt controller (GIC) in the PS. For more information refer to the *PS and PL based Ethernet in Zynq MPSoC wiki* [Ref 4].

For further details on IP cores, see the *AXI 1G/2.5G Ethernet Subsystem v7.0 Product Guide* (PG138) [Ref 1], *1G/2.5G Ethernet PCS/PMA or SGMII v16.0 LogiCORE IP Product Guide* (PG047) [Ref 2], and *AXI DMA v7.1 LogiCORE IP Product Guide* (PG021) [Ref 6].

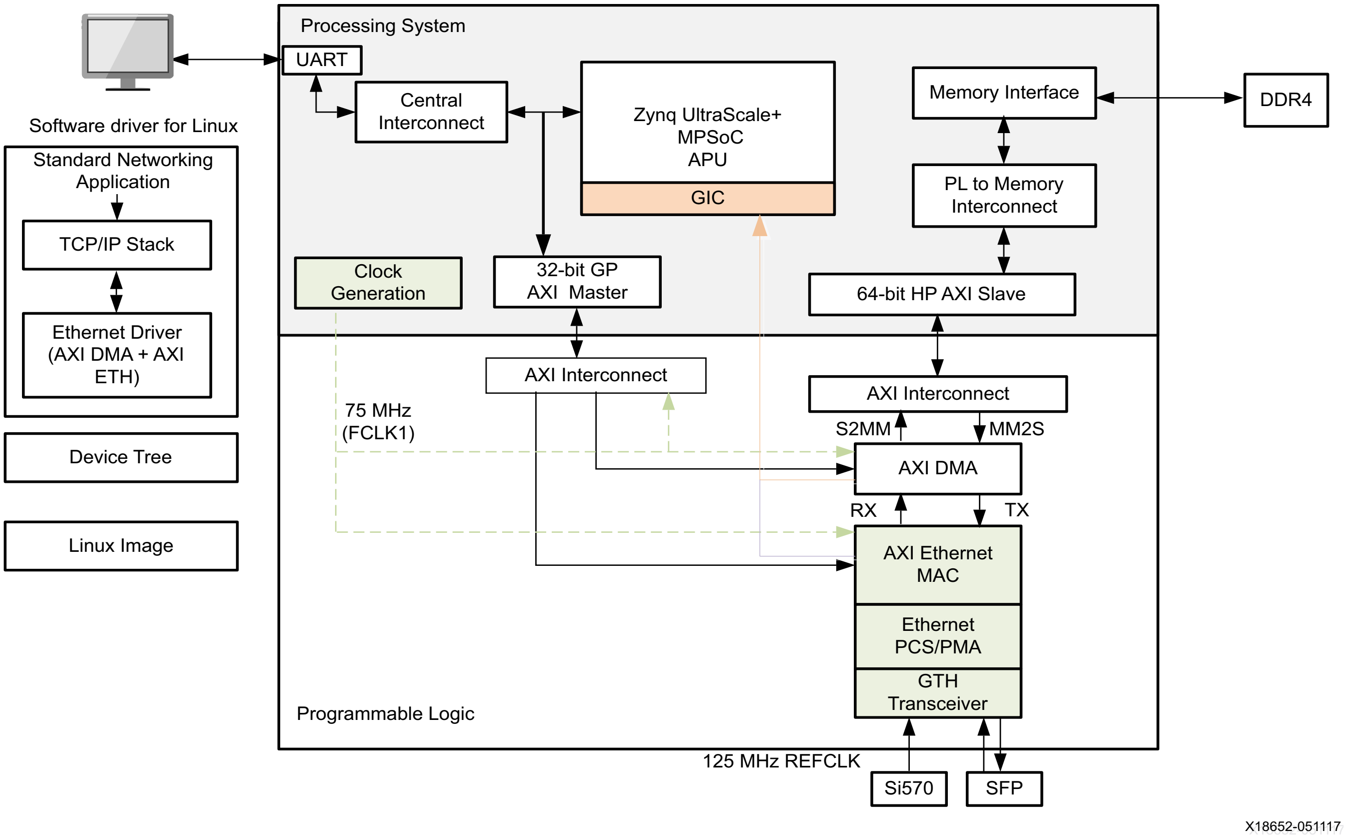

Figure 4: **1000BASE-X/SGMII PL Ethernet Design**

Reference Clock Generation

The GTH transceiver X0Y4 on the Zynq UltraScale+ MPSoC is connected to the SFP cage on the ZCU102 board for 1000BASE-X/SGMII transceivers. The GTH transceiver reference clock (125 MHz differential) is generated from the Si570 jitter attenuator on the ZCU102 board. The clock divider values are adjusted to generate 125 MHz from the Si570 programmable oscillator. The Si570 is programmed over the I2C interface to generate the required clock value. See the Si570 data sheet [Ref 5] for details on the Si570.

Software Design

This section describes the software aspects of the design. The monolithic Linux driver code facilitates the functionality listed below.

- PL Ethernet MAC accesses
- AXI DMA transfers
- Physical media initialization for 1000BASE-X interface using the phylib subsystem

Linux Driver

The software architecture for this design is shown in Figure 5. The driver is divided into the following sections.

- Initialization
- MAC driver hooks
- Interrupt service routines

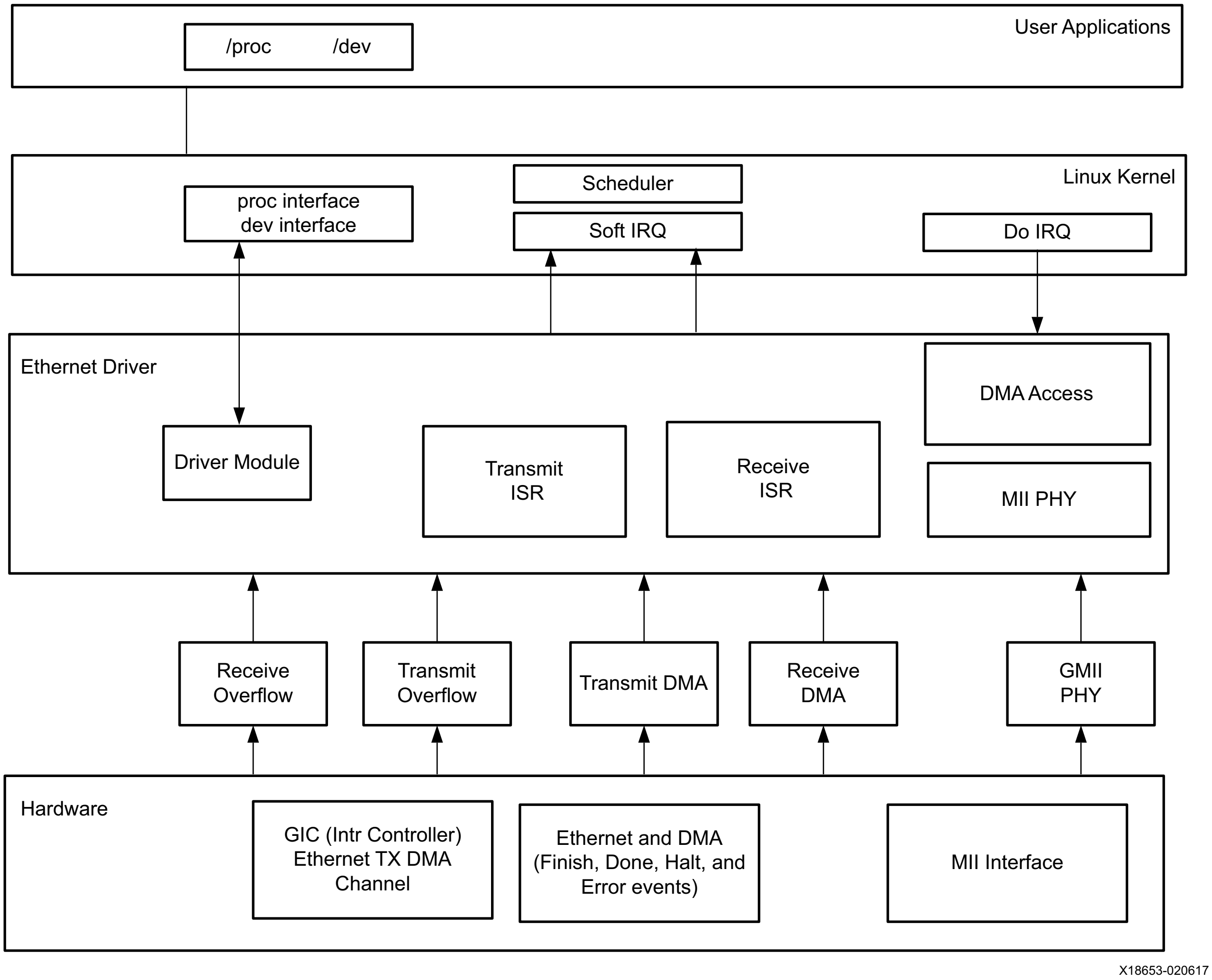

Figure 5: **Driver Architecture for PL Ethernet**

Using PL 10G Ethernet

This section describes a PL implementation of the 10G Ethernet. The design consists of 10G/25G high-speed Ethernet subsystem, AXI DMA, and AXI Interconnect IP cores. This design uses the high performance (HP) port for fast access to the PS-DDR memory. The general purpose slave port can also be used if the HP port is occupied with other peripherals.

Hardware Design

The 10G Ethernet implementation in PL is shown in Figure 6. The Xilinx 10G/25G high speed Ethernet subsystem implements the 25G Ethernet MAC with a physical coding sublayer (PCS) as specified by the 25G Ethernet Consortium. The 156.25 MHz reference clock to the transceiver is provided by the Si570 programmable oscillator available on the ZCU102 board. For more information refer to the *PS and PL based Ethernet in Zynq MPSoC wiki* [Ref 4].

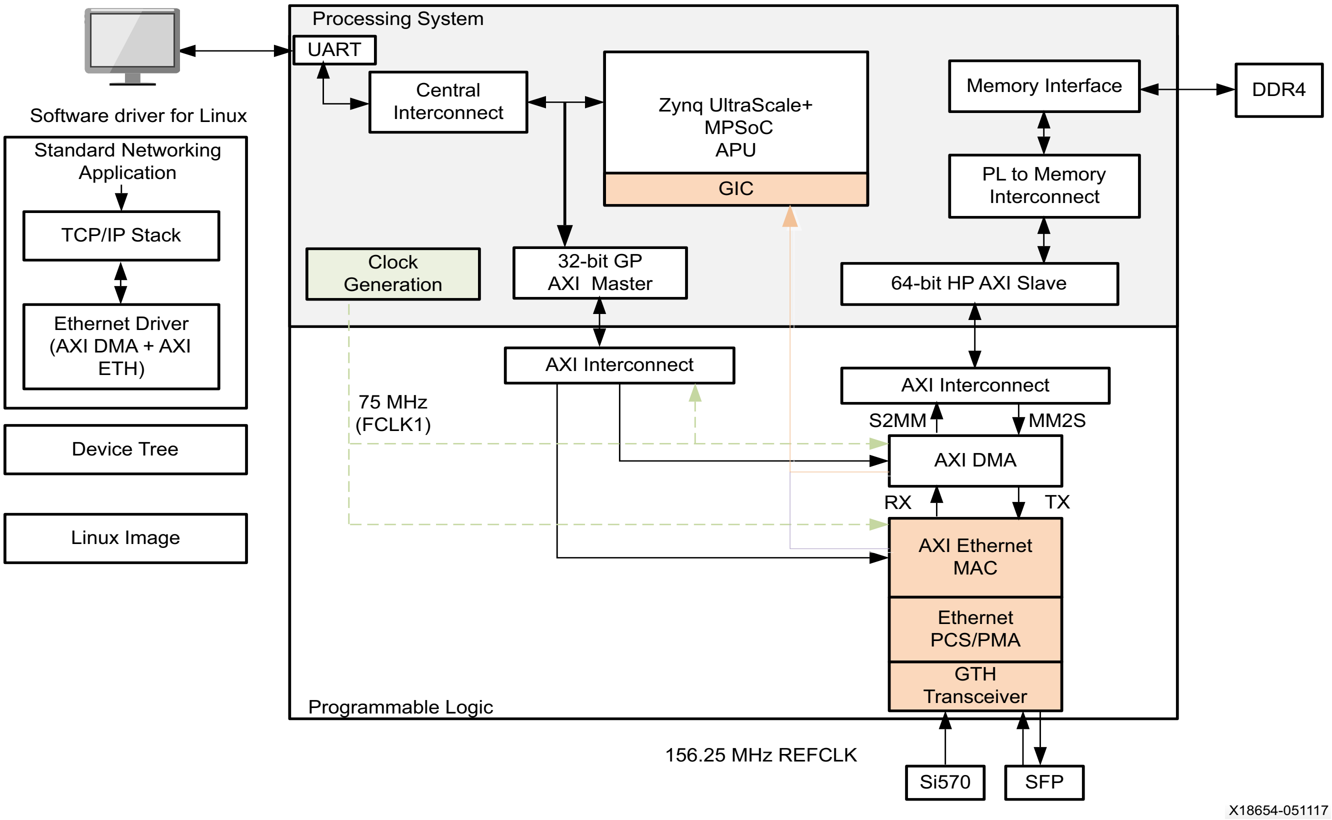

Figure 6: **10G PL Ethernet Design**

Reference Clock Generation

The GTH transceiver X0Y4 on the Zynq UltraScale+ MPSoC is connected to the SFP cage on the ZCU102 board. The GTH transceiver reference clock (156.25 MHz differential) is generated from the Si570 jitter attenuator on the ZCU102 board. The clock divider values are adjusted to generate 156.25 MHz from the Si570 programmable oscillator. The Si570 is programmed over the I2C interface to generate the required clock value. See the Si570 data sheet [Ref 5] for details on the Si570.

Software Design

This section describes the software aspects of the design. The monolithic Linux driver code facilitates the functionality listed below.

- PL Ethernet MAC accesses
- AXI DMA transfers

Linux Driver

The software architecture for this design is shown in Figure 5. The driver is divided into the following sections.

- Initialization
- MAC driver hooks
- Interrupt service routines

Device Trees

The device tree is a data structure for describing hardware. Instead of hard coding every detail of a device into an operating system, many aspects of the hardware can be described in a data structure that is passed on to the operating system at boot time. These settings are parsed by the drivers at the time of loading and the parameters are set as defined in the device tree. The Linux drivers' device trees consist of the following parameters.

- PS Ethernet MAC EMIO-specific:
  - PS GEM0 section contains the PS MAC parameters
- PL Ethernet-specific (1G/10G):
  - DMA section contains the AXI DMA parameters
  - Ethernet section contains the AXI Ethernet MAC parameters

Hardware and Software Requirements

The following hardware and software is required for testing the designs explained in this application note.

- Standard PC, running the Linux OS
- Ethernet port supporting 1000 Mb/s
- SFP for 1G and SFP+ for 10G modules
- 10G NIC for the host machine
- Netperf tool [Ref 7]
- iPerf tool [Ref 8]
- Vivado tools 2018.3 (IPI Design) [Ref 9]
- PetaLinux 2018.3 XSDK [Ref 10]
- Zynq UltraScale+ MPSoC ZCU102 board with an SFP-to-RJ45 adapter module for testing

The board setup for the 1G interface is shown in Figure 7. Jumper J16 should be shorted to enable transmission through the SFP. This design was tested with the Cisco GLC-T 1000BASE-X Ethernet to SFP Module.

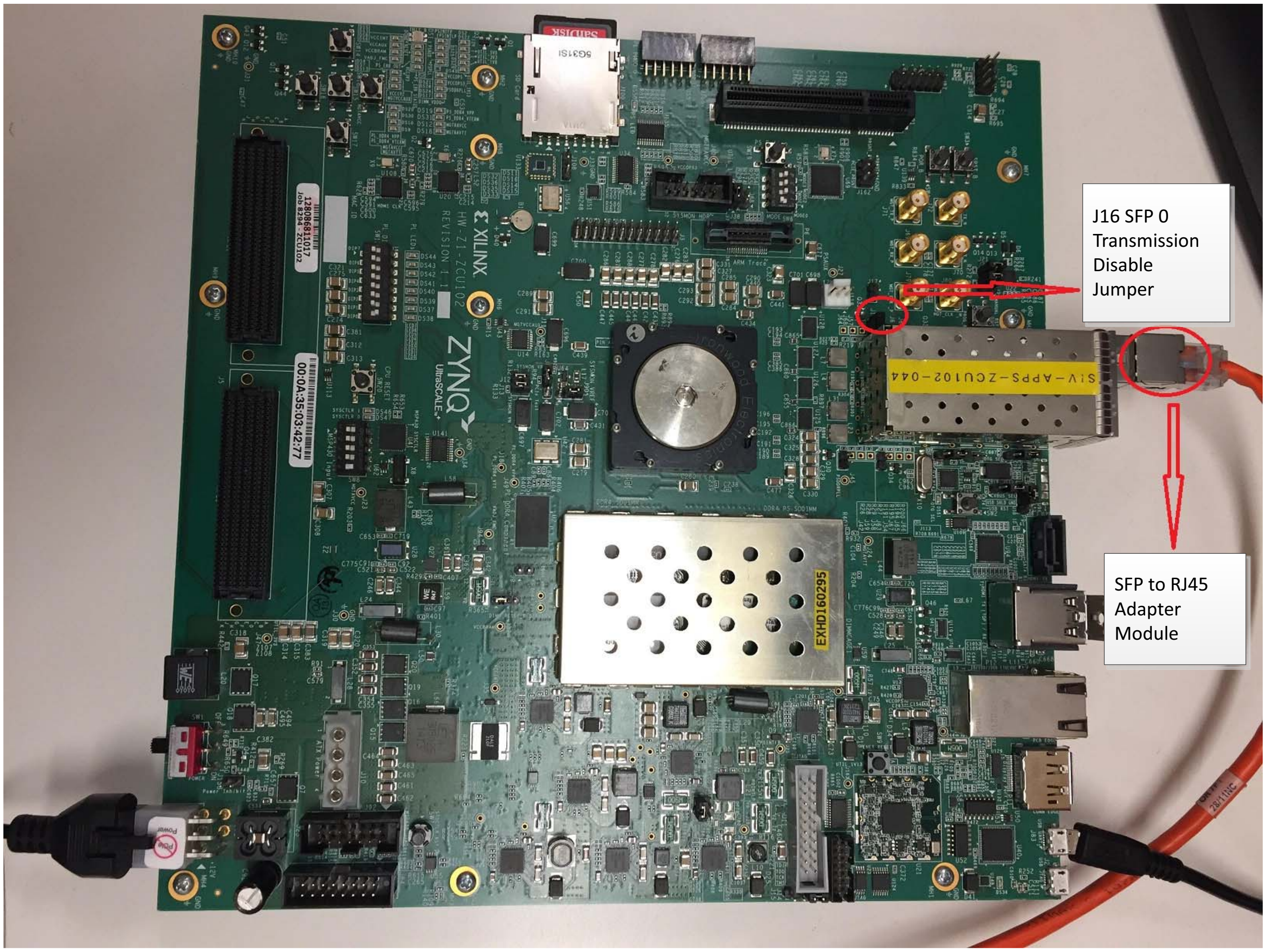

Figure 7: **Hardware Setup for 1G PL Ethernet**

The board setup for the 10G interface is shown in Figure 8. Jumper J16 should be shorted to enable transmission through the SFP. This design was tested with the Avago afbr-709smz optical to Ethernet SFP+ module.

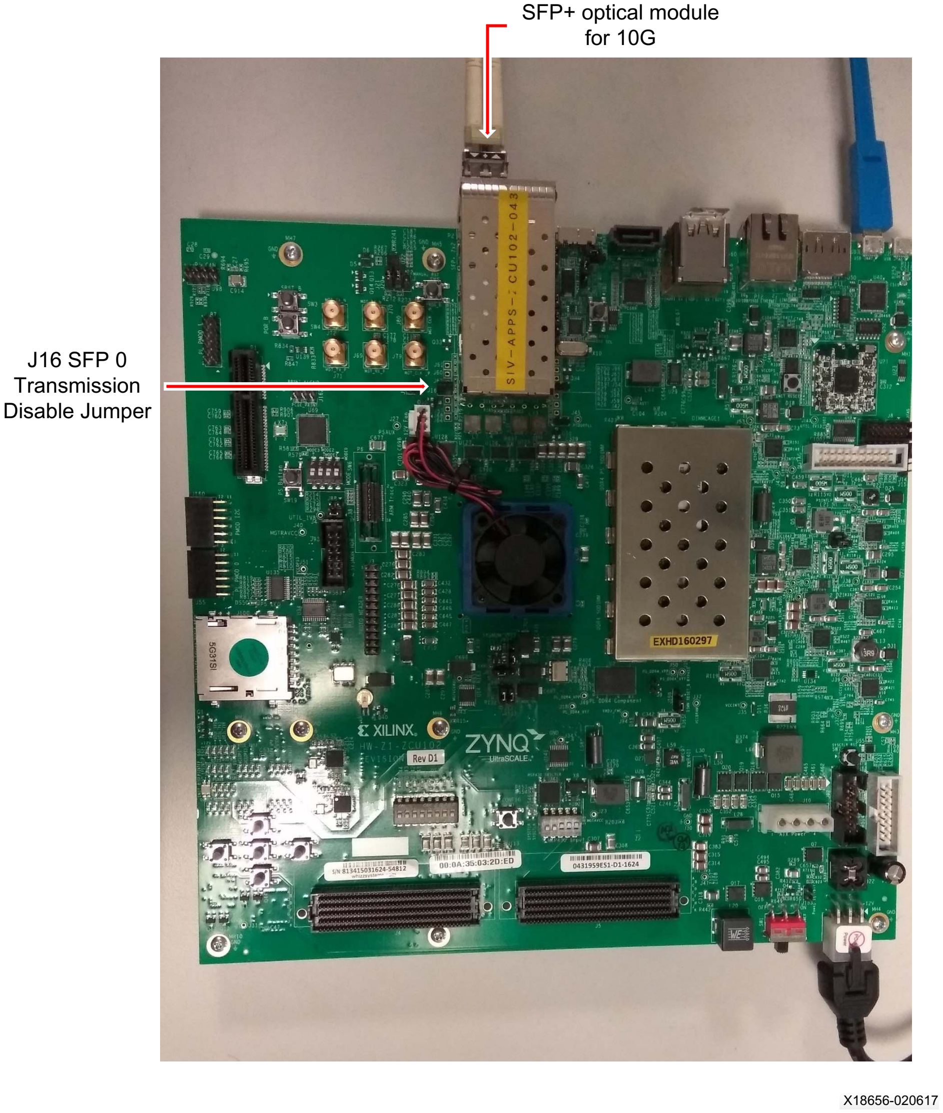
SFP+ optical module for 10G
J16 SFP 0 Transmission Disable Jumper
X18656-020617

Figure 8: **Hardware Setup for 10G PL Ethernet**

Conclusion

This application note describes various Ethernet based implementations. The performance benchmarking results for the designs included in this application note can be found in the *PS and PL based Ethernet in Zynq MPSoC wiki* [Ref 4].

Reference Design

Download the reference design files for this application note from the Xilinx website.

Refer to the instructions in the readme for building the hardware and software code.

Table 1 shows the reference design matrix.

Table 1: **Reference Design Matrix**

Parameter	Description
General	
Developer names	Bhargav Shah, Naveen Kumar Gaddipati, Akhilesh Mahajan, and Sutej Kulkarni
Target devices	Zynq UltraScale+ Devices
Source code provided	Yes
Source code format	Verilog, C
Design uses code and IP from existing Xilinx application note and reference designs or third party	Yes
Simulation	
Functional simulation performed	No
Timing simulation performed	No
Test bench used for functional and timing simulations	No
Test bench format	N/A
Simulator software/version used	N/A
SPICE/IBIS simulations	N/A
Implementation	
Synthesis software tools/versions used	Vivado tools 2018.3 PetaLinux 2018.3 XSDK
Implementation software tools/versions used	Vivado tools 2018.3 PetaLinux 2018.3 XSDK
Static timing analysis performed	Yes
Hardware Verification	
Hardware verified	Yes
Hardware platform used for verification	ZCU102 evaluation board

References

1. *AXI 1G/2.5G Ethernet Subsystem v7.0 Product Guide* (PG138)
2. *1G/2.5G Ethernet PCS/PMA or SGMII v16.0 LogiCORE IP Product Guide* (PG047)
3. *10G/25G High Speed Ethernet Subsystem v2.0 Product Guide* (PG210)
4. PS and PL based Ethernet in Zynq MPSoC
5. *Si570 Data Sheet* (www.silabs.com/Support%20Documents/TechnicalDocs/Si570.pdf)
6. *AXI DMA v7.1 LogiCORE IP Product Guide* (PG021)
7. Netperf (www.netperf.org)
8. iPerf (http://sourceforge.net/projects/iperf/)
9. Xilinx Vivado Design Suite
10. PetaLinux
11. *Zynq UltraScale+ MPSoC Technical Reference Manual* (UG1085)
12. *ZCU102 Evaluation Board User Guide* (UG1182)
13. *UltraScale Architecture GTH Transceivers User Guide* (UG576)

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
04/30/2019	1.4	Updated for Vivado Design Suite 2018.3 and PetaLinux tools 2018.3 release.
10/10/2018	1.3	Updated for Vivado Design Suite 2018.1 and PetaLinux tools 2018.1 release.
05/10/2018	1.2	Updated for Vivado Design Suite 2017.3 and PetaLinux tools 2017.3 release.
09/20/2017	1.1	Updated for Vivado Design Suite 2017.1. Corrected transceiver references from GTX to GTH transceivers.
03/24/2017	1.0	Initial Xilinx release.

Please Read: Important Legal Notices

The information disclosed to you hereunder (the “Materials”) is provided solely for the selection and use of Xilinx products. To the maximum extent permitted by applicable law: (1) Materials are made available "AS IS" and with all faults, Xilinx hereby DISCLAIMS ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE; and (2) Xilinx shall not be liable (whether in contract or tort, including negligence, or under any other theory of liability) for any loss or damage of any kind or nature related to, arising under, or in connection with, the Materials (including your use of the Materials), including for any direct, indirect, special, incidental, or consequential loss or damage (including loss of data, profits, goodwill, or any type of loss or damage suffered as a result of any action brought by a third party) even if such damage or loss was reasonably foreseeable or Xilinx had been advised of the possibility of the same. Xilinx assumes no obligation to correct any errors contained in the Materials or to notify you of updates to the Materials or to product specifications. You may not reproduce, modify, distribute, or publicly display the Materials without prior written consent. Certain products are subject to the terms and conditions of Xilinx’s limited warranty, please refer to Xilinx’s Terms of Sale which can be viewed at http://www.xilinx.com/legal.htm#tos; IP cores may be subject to warranty and support terms contained in a license issued to you by Xilinx. Xilinx products are not designed or intended to be fail-safe or for use in any application requiring fail-safe performance; you assume sole risk and liability for use of Xilinx products in such critical applications, please refer to Xilinx’s Terms of Sale which can be viewed at http://www.xilinx.com/legal.htm#tos.

AUTOMOTIVE APPLICATIONS DISCLAIMER

AUTOMOTIVE PRODUCTS (IDENTIFIED AS "XA" IN THE PART NUMBER) ARE NOT WARRANTED FOR USE IN THE DEPLOYMENT OF AIRBAGS OR FOR USE IN APPLICATIONS THAT AFFECT CONTROL OF A VEHICLE ("SAFETY APPLICATION") UNLESS THERE IS A SAFETY CONCEPT OR REDUNDANCY FEATURE CONSISTENT WITH THE ISO 26262 AUTOMOTIVE SAFETY STANDARD ("SAFETY DESIGN"). CUSTOMER SHALL, PRIOR TO USING OR DISTRIBUTING ANY SYSTEMS THAT INCORPORATE PRODUCTS, THOROUGHLY TEST SUCH SYSTEMS FOR SAFETY PURPOSES. USE OF PRODUCTS IN A SAFETY APPLICATION WITHOUT A SAFETY DESIGN IS FULLY AT THE RISK OF CUSTOMER, SUBJECT ONLY TO APPLICABLE LAWS AND REGULATIONS GOVERNING LIMITATIONS ON PRODUCT LIABILITY.

© Copyright 2017–2019 Xilinx, Inc. Xilinx, the Xilinx logo, Artix, ISE, Kintex, Spartan, Virtex, Vivado, Zynq, and other designated brands included herein are trademarks of Xilinx in the United States and other countries. AMBA, AMBA Designer, ARM, ARM1176JZ-S, CoreSight, Cortex, PrimeCell, and MPCore are trademarks of ARM in the EU and other countries. All other trademarks are the property of their respective owners.