

BRX-WP002: Non-intrusive continuous multi-gigabit transceivers link monitoring

White Paper

Table of Contents

1 History.....	3
2 Introduction.....	3
3 Link monitoring.....	6
4 References.....	9

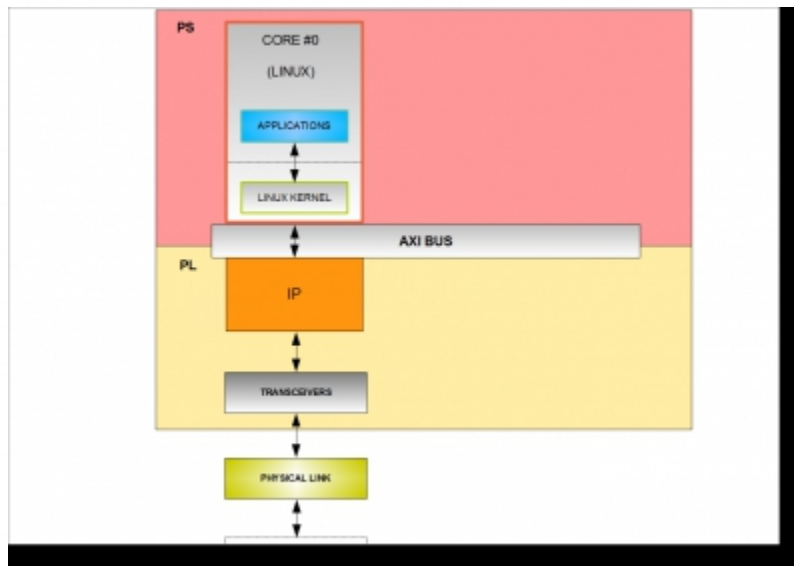
1 History

Version	Date	Notes
1.0.0	June 2016	First Public Release

2 Introduction

This White Paper describes a practical application of the asymmetric multi-processing (AMP) configuration illustrated [here](#) [a]. Specifically, this approach has been used to implement a non-intrusive continuous link monitoring mechanism for the Xilinx Zynq multi-gigabit serial transceivers [b]. The starting point for this work is represented by the Xilinx Application Notes XAPP743[1] and XAPP1198[2], thus reading of these documents is highly recommended.

The issue that this White Paper addresses is the need of monitoring multi-gigabit transceivers link status when operating on the field [c] in a non-intrusive way. Just imagine a product that is based on the architecture similar to the one depicted in the following figure. It is assumed that this architecture is quite representative of many real use cases.



Concept block diagram of the system without monitoring subsystem

A generic communication IP is implemented in Programmable Logic (PL) . This IP makes use of multi-gigabit transceivers to communicate with the peer at the other end of the physical link. On Processor Subsystem (PS) side, Linux operating system is used. On top of the kernel, several applications run, implementing high-level product's functionalities, including the management of sent and received data through the link shown in the figure. It is also assumed that the reliability of this link is a crucial factor for the successful product functioning. Thus a specific monitoring of its health has to be implemented in order to detect any deviation from normal working conditions that may affect link robustness such as:

- medium/long-term drift of the physical link characteristics
- significant part to part variations of such characteristics.

Last but not least, monitoring functionality has to be

substantially non-intrusive with respect to:

- REQ1: software applications running in the Linux realm
- REQ2: user functions implemented in PL
- REQ3: transmission/reception of user data through the link
- REQ4: the overall system functionality.

[a] At the time of this writing, [this document](#) refers to BORA only. However the same AMP implementation is available for BORA Xpress as well.

[b] More complex configurations such as [this one](#) can be used as well.

[c] That is after the product has been deployed.

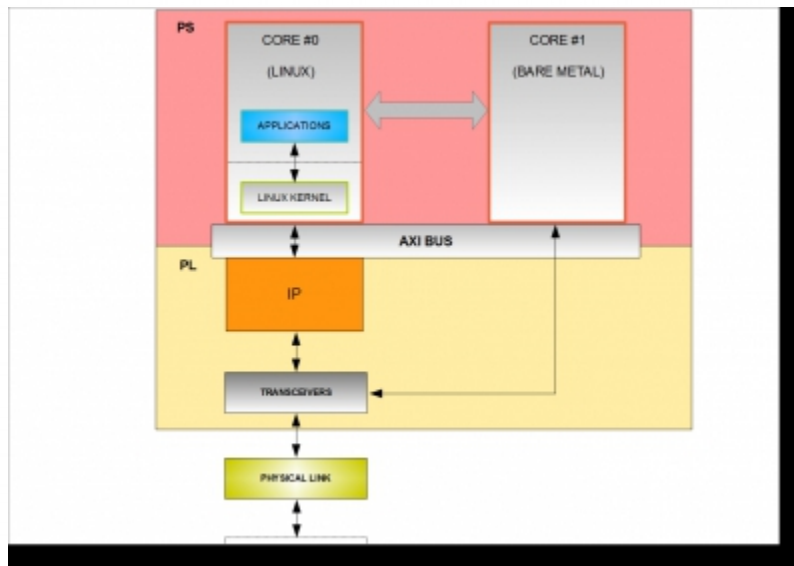
3 Link monitoring

Before illustrating a possible solution, some further considerations about requirements listed in previous section are illustrated.

[REQ1](#) through [REQ4](#) specify in detail a generic requirement of non-intrusiveness. In other words, it is required that the implementation of link monitoring function is virtually transparent to the other parts of the system. As a consequence

- application software don't need any change when adding such functionality
- the cost of additional PL resources is neglectable with respect to the Microblaze-based approach described in [\[1\]](#)
- physical link is not affected at all: normal operating, while monitoring is continuously in progress, is guaranteed by the integrated circuitry of GTP/GTX transceivers[\[3\]](#)[\[4\]](#).

Block diagram of the solution here described is depicted in the following figure.



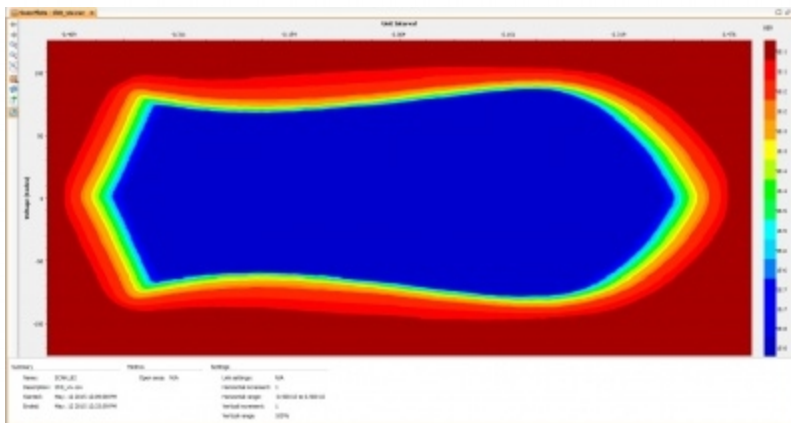
Concept block diagram of the system with monitoring subsystem

Conceptually, the system architecture amounts to the one illustrated in [1] with two notable differences:

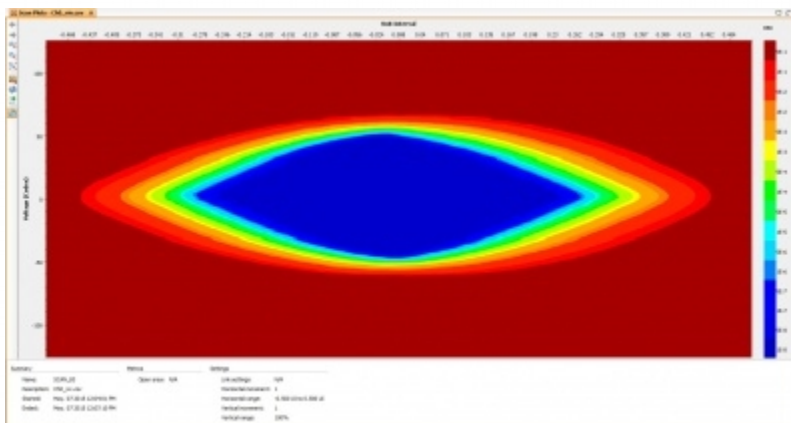
- since monitoring software is executed by the second Cortex A9 core, MicroBlaze infrastructure - that would consume PL resources - is not necessary
- thanks to the communication channel between the two ARM cores, core #0 can be signalled by core #1 in case an alert condition is detected at the transceivers level. Thus appropriate actions can be taken at application level.

This solution has been tested on BORA Xpress/BORA Xpress EVB platform implementing PCIe connectivity. PCIe Root Complex and AXI-to-DRP bridge have been integrated in PL. JTAG-to-AXI bridge has been included as well in order to keep the possibility to access monitoring data via JTAG during development/debugging stage. This allows to retrieve data from

host running Vivado and generate 2D statistical eye diagrams very easily (for more details please refer to [1]). The following figures show a couple of such diagrams.



PCIe Gen1 (2.5 Gbps) statistical eye diagram



PCIe Gen2 (5.0 Gbps) statistical eye diagram

4 References

1. Mike Jenkins, David Mahashin, XAPP743 (v1.0.1) Eye Scan with MicroBlaze Processor MCS, 28th October 2013
2. Luis Bielich, XAPP1198 (v1.1) In-System Eye Scan of a PCI Express Link with Vivado IP Integrator and AXI4, 19th November 2014
3. UG482 7 Series FPGAs GTP Transceivers User Guide, http://www.xilinx.com/support/documentation/user_guides/ug482_7Series_GTP_Transceivers.pdf
4. UG476 7 Series FPGAs GTX/GTH Transceivers User Guide, http://www.xilinx.com/support/documentation/user_guides/ug476_7Series_Transceivers.pdf