2013 Eighth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing

Benchmarking of PCIe-performance in microTCA-equipment

K. Lostrie*, P. De Meulenaere*, M. Temmerman*, N. Van Remortel†, W. Beaumont†
*CoSys-Lab, University of Antwerp
Paardenmarkt 92, 2000 Antwerpen, Belgium
{koen.lostrie, paul.demeulenaere, marijn.temmerman}@uantwerpen.be
†EDF, University of Antwerp
Groenenborgerlaan 171, 2020 Antwerpen, Belgium
{nick.vanremortel, wim.beaumont}@uantwerpen.be

***Abstract*—In the development of advanced data acquisition, testing or telecommunication equipment, one often relies on modular off-the-shelf processor boards and I/O-boards that are being composed to a single distributed system. To support such architectures, the microTCA (micro Telecommunication Computing Architecture) offers standardized racks and standardized backpanel communication protocols connecting the different boards.**

During the design of such distributed systems, the performance characteristics of the end-to-end communication between different boards in the system are often not sufficiently known. In this paper, we present a setup allowing for benchmarking the performance of the full PCIe-communication path between two microTCA FPGA-boards. The experimental setup is discussed, and it is shown that benchmarking figures for throughput, delay and delay variation of the PCIe end-to-end path can be retrieved.

I. Introduction

The design of complex data acquisition (DAQ) instruments is often tailored to a specific application. For instance, the detectors at the elementary particle accelerators in CERN produce huge amounts of data that need to be filtered instantaneously in the DAQ-equipment such that only the physically interesting data is being sent to central computing equipment. A single card in such system typically receives data over 96 analogue optical fibers which, after digitization, represent a total input rate of 3.4 GByte/s. Such front-end cards need to reduce these data by applying algorithms that select interesting events only, resulting in an average data output in the order of 200 MByte/s [1].

Due to the demanding processing requirements in such systems, these embedded data acquisition units will need distributed processing with instant data exchange between the processing units. To limit cost and effort, one typically looks for off-the-shelf hardware solutions that provide sufficient flexibility to compose a system with the required functionality, and with the possibility to add own software and FPGA-code. One of the candidate platforms is the microTCA [2], which is an industrial standard that describes a small crate and backpanel. Several industrial implementations are available, as well as an extensive number of cards that can be plugged into the backpanel, hence to compose a single system. This way the microTCA ecosystem, although original defined for telecommunication equipment, is very well suited to build advanced DAQ- and testing equipment.

The microTCA-backplane contains several options for communication between the connected cards: 10 Gigabit Ethernet, PCI-express (PCIe) or Serial Rapid IO, all having performance characteristics in the same order of magnitude. One of the non-obvious questions to be answered during the system design of a DAQ is therefore which backplane communication technology to select for the envisaged system. This choice will not only influence the final performance of the system, but will also determine the communication technology that needs to be present on the cards to be inserted in the system.

Besides the crate and backpanel, the microTCA defines a number of cards, where the MCH (MicroTCA Carrier Hub) and the AMC (Advances Mezzanine Card) are the most important ones. Next to some other central tasks, the MCH contains different switches and fabrics allowing the AMCs to communicate. The AMC provide interfaces at the front, e.g. for I/O purposes in a DAQ, and often contain processors or FPGAs.

In this paper, we present the benchmarking setup and measurements of the microTCA PCIe backplane properties, with focus on the end-to-end path, i.e. the path between the user logic on one card and the user logic on another card. This link entails more than just the PCIe-bus. It also includes internal buses on the card, the effect of the PCIe-bridge, etc. The obtained characteristics can be used as input for making the right design decisions during the further design of the distributed data processing application for the DAQ. It will contribute to the estimation whether the real-time constraints can be met by the algorithms, and it will allow for estimating buffer sizes for the backplane traffic. Moreover, the communication characteristics will potentially trigger a redesign of the algorithms, e.g. in terms of the distribution of the algorithm over different processing cores.

Our goal to benchmark the PCIe-performance of a given hardware architecture resembles to some extent the results described in [3], [4] where the PCIe-performance is evaluated for the InfiniBand architecture. Also in this case, the authors measure the end-to-end throughput and delay, in which a PCIe-link is included. The benchmark test patterns are quite straightforward - they call it 'microbenchmarks' - and they claim to design more application-level testbenches to better approach realistic situations.

978-0-7695-5094-7/13 $31.00 © 2013 IEEE
DOI 10.1109/3PGCIC.2013.81

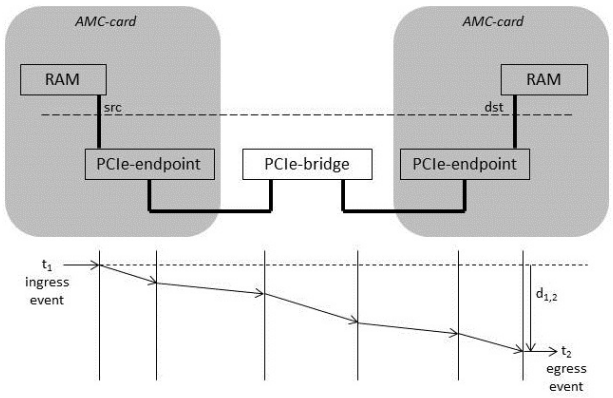


Fig. 1. Definition of the end-to-end path and of packet transfer delay events

The remainder of this paper is organized as follows. The next section briefly discusses the definitions we use on bandwidth, latency and latency variation. Next, the experimental setup is described. In section IV, we describe the experimental results that are discussed in a final section.

II. Definitions

As mentioned before, the data of interest in this benchmarking are: the throughput of the bus, the average delay of a packet, and the variation of the packet delay. With the term 'packet', we denote here an amount of data that is sent contiguously over the bus. In the scope of the current paper, a packet corresponds to the length of one DMA-transfer, as will be motivated in the experimental setup.

To define the throughput, delay and delay variation exactly, we refer to figure 1. In the present setup, packets are sent from the RAM of one AMC-card to the RAM of another AMC-card. The total transfer path includes internal buses, the PCIe-bus, and several components. What is of most interest to the application designer, are the communication properties of the complete end-to-end link, i.e. between the points indicated with 'src' and 'dst', without the need for information on the individual sublinks of this path. The present benchmarking setup will only cover the span from src to dst.

The guidelines of ITU-T Y.1540 [5] define throughput, packet transfer delay and packet delay variation for IP-packets. In the present case, we however do not deal with IP-packets but with DMA-transfers. To align as close as possible to mentioned ITU-T definitions, we introduce the following similar definitions.

- The *throughput* is the total number of bits of a packet (or, DMA-transfer) that were successfully transmitted at a given egress point during a specified time interval divided by the time interval duration. Its unit is bit/s. It is also often referred to as *bandwidth*, or *busspeed*.
- The *packet transfer delay* is the time, $t_2 - t_1$ between two reference events, being the ingress event of the first bit of the packet at time t_1 at the source src and the egress event of the last bit at time t_2 at the destination dest. Its unit is s. It is often shortened as *delay*.

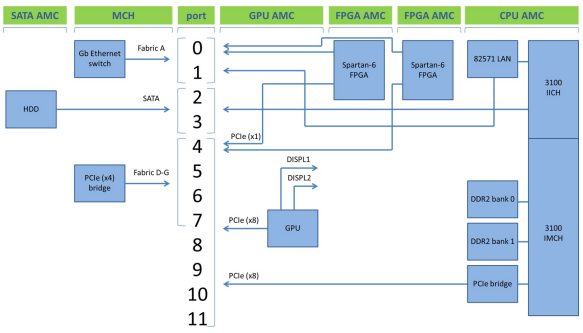


Fig. 2. Overview of the slot filling and backplane connectivity

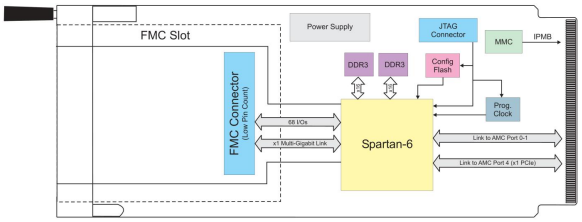


Fig. 3. Block diagram of the TAMC631 FPGA-card [7]

- The *packet delay variation* v_k for an individual packet k between src and dst is the difference between the absolute packet transfer delay x_k of the packet and a defined reference packet transfer delay, $d_{1,2}$, between those same src and dst points:

$$v_k = x_k - d_{1,2}$$

  As for the reference packet delay, we will use in this paper the average packet transfer delay. Its unit is s, and it is often shortened as *delay variation*.

Notice that the packet delay variation refers to a single packet. To achieve some idea of the statistical distribution of the packet delay variations, the set of measurements can be divided into quantiles. However, in this paper, we will limit ourselves to pointing to the maximum delay variation observed in the experiments.

III. Experiment setup

A. System architecture

For the experiments discussed in this paper, we use a microTCA-setup starting from the NAT-START2-PLUS basic setup, which is based on the NATIVE-SX crate. In the available slots of this crate two Tews TAMC631-12R AMC-cards are inserted, each containing a Xilinx Spartan-6 FPGA. Figure 2 shows an overview of the setup. The data path under study in this paper starts at the FPGA on one AMC. Via the PCIe-link, it reaches the PCIe-bridge on the MCH. From this bridge, a second PCIe-link makes the connection to another FPGA on a second AMC.

Figure 3 shows a block diagram of the TAMC631 AMC-card, where it can be seen that the Spartan-6 has a PCIe-link to the backplane. At the front of the card, an FMC (FPGA Mezzanine Card) connector is available. We use this front

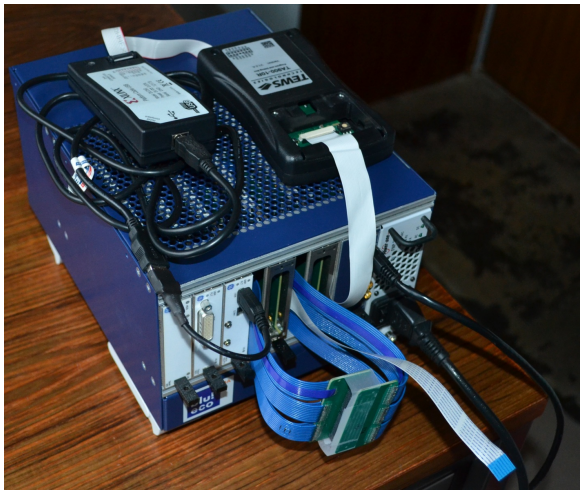


Fig. 4. The NATIVE-SX crate with TAMC631 FPGA-cards

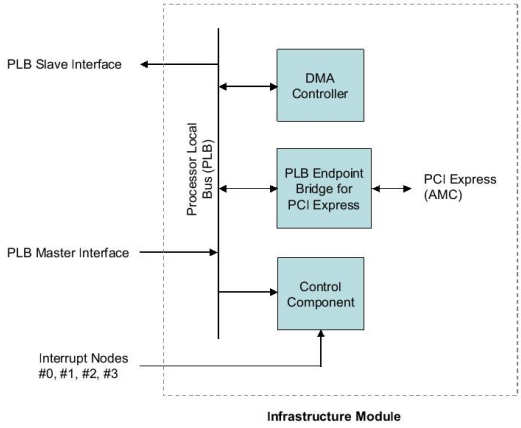


Fig. 5. The TAMC631 reference design: Infrastructure Module [6]

connector to distribute a start signal between both boards, indicating accurately the start of the experiments. This way, measurements at both sides are being synchronized. Figure 4 shows the global microTCA-crate setup, where the blue FMC-cables in the front are clearly visible.

B. System design

The reference design included with the TAMC631-cards contains IP-blocks for the connectivity to the PCIe, for the DMA-controller, and a User Logic Area where the Register Set block contains user-definable registers, as shown in figures 5 and 6.

The end-to-end traffic path starts at the DMA-controller, fetching data from memory according to settings configured in the Register Set. These data are transferred over the internal Processor Local Bus (PLB) to the bridge that connects the PLB to the PCIe-lane. The second AMC-card subsequently receives these packets over the identical data path, until the respective DMA-controller writes the data to memory.

Figure 7 shows an overview of the changes performed in

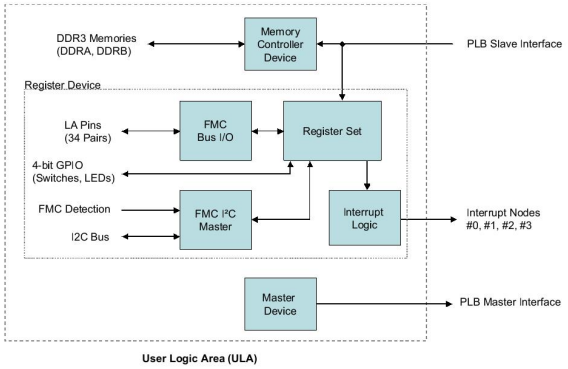


Fig. 6. The TAMC631 reference design: User Logic Area [6]

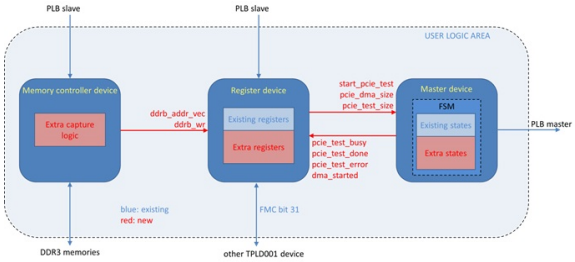


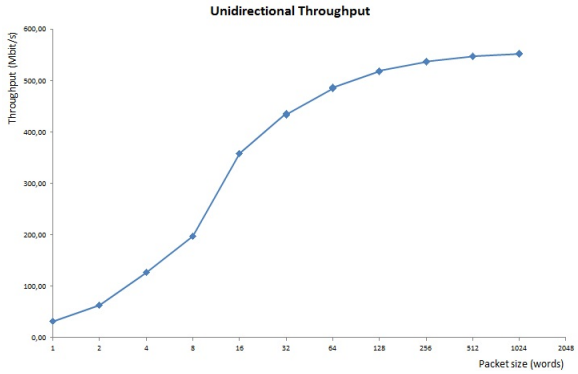
Fig. 7. Summary of the changes done in the User Logic Area

the User Logic Area to setup the experiments. The essence of the experiment design can be found in the Register Set, where registers have been defined to store the configuration settings and the measurement data, such as the start of the lengths of the DMA-packets at sender and receiver side, the timestamps corresponding to the end of sending or receiving the data, etc. The Master Device in the User Logic Area contains a finite state machine that governs the flow control of the program. It is altered to steer the current experiments; e.g. the synchronization signal over the FMC-bus is controlled in this part. Finally, in the Memory Controller Device, some extra logic has been added to record the required addresses on the PLB-bus in a correct way.

C. Performed experiments

In order to characterize the end-to-end throughput, delay and delay variation, a number of experiments are executed. The strategy used is first to characterize unidirectional traffic only. This way, a number of 'pure' characteristics of the end-to-end path become apparent, such as the optimal throughput, effect of packet overhead, etc. Different measurements are taken for different DMA-packet lengths, varying from 1 word up to 1024 words (1 word or 32 bits corresponds to the width of the PLB-bus). A bunch of 10 runs is executed per DMA-packet length. In each run of the experiment, the size of the complete data transfer is kept constant to 16k words.

To measure the characteristics for bidirectional traffic, the experiment is conceived analogous to the unidirectional measurements, but while this unidirectional traffic is ongoing, different kinds of traffic flows are superimposed in the opposite traffic direction. Since there are several congestion points



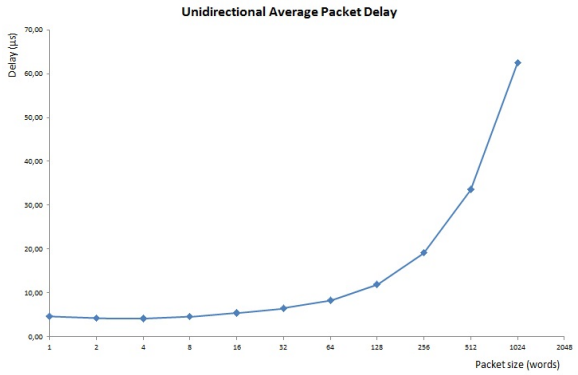
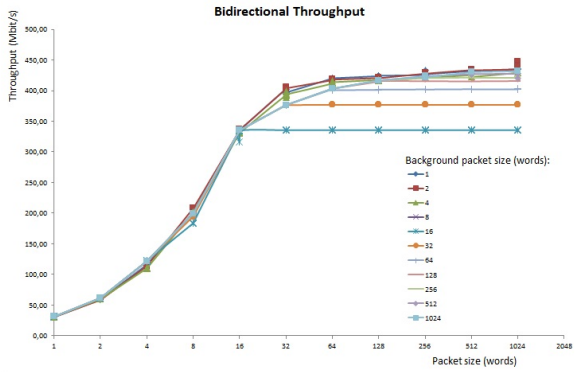


Fig. 8. Throughput and packet delay for unidirectional traffic



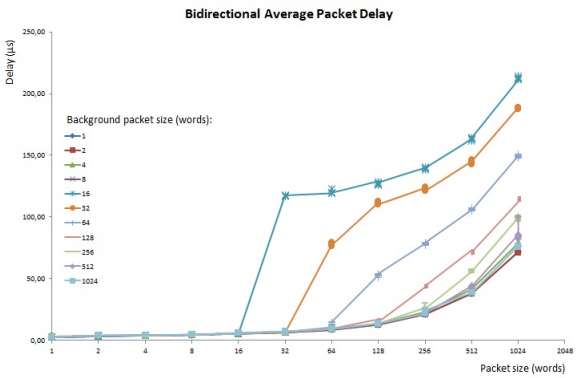


Fig. 9. Throughput and packet delay for bidirectional traffic against different profiles for the background traffic

possible in the end-to-end path, such as the PLB-bus or the PCIe-bridge, this 'background traffic' will impose its influence on the original unidirectional traffic. In the congestion points, scheduling rules of the bus will determine which packets are allowed to pass, and which need to wait. Therefore, not only the amount of background traffic, but also its packet size will influence the traffic characteristics of the monitored traffic flow. To observe the potential effects of the packet size of the background traffic on the original traffic, the experiments from the unidirectional traffic are repeated against background traffic with varying DMA-packet lengths from 1 word up to 1024 words.

For all experiments, the clock rate of the PLB-bus is configured to 62.5 MHz.

IV. Measurements

Figure 8 summarizes the measurements for the unidirectional traffic. Its throughput clearly depends on the DMA-packet length. This is most probably due to the overhead that needs to be included for each packet to be transmitted: the smaller the packet, the larger the relative overhead. The average packet delay depends on the packet size as well. Smaller packets obviously need smaller delays for their total transmission compared to large packets.

Notice that the full PCIe-throughput has not been reached in these experiments, which is due to fixed settings in the FPGA reference design used in the current experiments; these settings have however no impact on the validity of the proposed benchmarking strategy.

The results for the bidirectional traffic have been summarized in figure 9. The total throughput is in all cases smaller compared to the unidirectional traffic. Depending on the packet size of the background traffic, it can be observed that a background containing either small or large packets still returns reasonably high throughput figures. Background traffic containing packets of 16 or 32 bytes apparently inhibit the ongoing traffic the most.

A similar trend can be observed for the average packet delay for bidirectional traffic. In addition, the measured delays are substantially higher compared to the case of the unidirectional traffic.

The observation of packet delay variations is much more complex. Measurement and statistical processing of all individual packets in the current experiment setup are not possible, due to limitations on the number of available registers. Nevertheless, the observations of the average packet delay already show large variations, which can serve as first estimates on the possible delay variation. This can be observed both in the unidirectional and bidirectional traffic experiments.

V. Discussion

The retrieved data for the throughput and packet delay characterize the end-to-end path containing a PCIe-link in the microTCA-setup. Due to the chosen measurement reference points at source and destination, the complete path includes the PLB-bus and bridge components on the FPGA-cards.

The recorded measurements can therefore be claimed for the given hardware setup and the given configuration settings. In case of different configurations (e.g. regarding the number of PCIe-lanes) or different hardware (e.g. usage of the 10 Gigabit Ethernet links), the benchmarking obviously needs to be redone.

Although the benchmarking traffic characteristics intend to emulate a broad range of realistic traffic characteristics, the validity of this benchmarking also has its limits. An appealing method would be to repeat the benchmarking with traffic that resembles the realistic traffic for a given application field. For instance, the design of DAQ-equipment in the CERN-experiments would necessitate benchmarking for distributed real-time pattern recognition algorithms. Standardized algorithms are not yet available to generate realistic traffic characteristics that can be used as a reference benchmark.

In [3], the authors use the word microbenchmarking' for their experiments, probably referring to the limited complexity of the test patterns they run. The test patterns presented in our work, have a similar, limited complexity. Same as these authors, we believe that application-level benchmarking is the next step to be taken to achieve fine-tuned figures for the PCIe-throughput, delay and delay variation.

Once trustworthy benchmarking results have been achieved, they can be used as calibrated traffic characteristics in the design of new distributed algorithms that need to run on the benchmarked microTCA-configuration. While developing such new applications, the effects of throughput, delay and delay variation will typically be used in the verification of the design by simulation. In the application design, the benchmark of the communication path will have its effect on: specific distribution of the software components over the different processing entities, optimization or parallellization of algorithms where necessary, design of higher data communication layers to improve efficiency and reliability, dimensioning of data buffers, etc.

VI. Conclusion

The present work presents a benchmarking setup for communication over PCIe in a microTCA-configuration. It approaches the problem from an application point of view, leading to the observation that not only the PCIe-bus needs to be benchmarked, but the complete communication end-to-end path, including internal buses.

The setup for the benchmarking experiments has been discussed, focusing to the possibilities and limitations of the Spartan-6 reference design in the TAMC631-cards that has been taken as the starting point for the benchmark. Figures for throughput and packet delay could be retrieved, as well as representative samples for packet delay variation.

It has been argued that the benchmarking results may contribute to application development. To achieve benchmarking results in realistic traffic environments, reference distributed algorithms need to be developed, which are representative for a specific application area, such as the high-speed real-time signal processing in DAQ-equipment.

References

[1] Foudas, C., Bainbridge, R. et al. The CMS tracker readout front end driver *IEEE Transactions on Nuclear Science*, 52(6): 2836–2840, Dec 2005.

[2] PICMG. Micro telecommunications computing architecture short form specification, Sep 2006.

[3] J. Liu, A. Mamidala, A. Vishnu, and D.K. Panda. Evaluating InfiniBand performance with PCI express. *IEEE Micro*, 25(1):20–29, Jan 2005.

[4] M. J. Koop, W. Huang, K. Gopalakrishnan, and D. K. Panda. Performance analysis and evaluation of PCIe 2.0 and quad-data rate infiniband. In *16th IEEE Symposium on High Performance Interconnects. HOTI 08.*, pages 85–92, Aug 2008.

[5] ITU-T Recommendation Y.1540. Internet protocol data communication service - IP packet transfer and availability performance parameters, Dec 2002.

[6] TEWS Technologies. TPLD001 - TAMC631 Platform Example Application Version 1.0 Design Documentation Issue 1.0.1 Apr 2011.

[7] TEWS Technologies. TAMC631 Spartan6 AMC with FMC Module Slot. Online: www.powerbridge.de/download/data/amc/TAMC631_DS_1106.pdf, consulted June 2013.