Improved performance for the ATLAS ReadOut System with the switchbased architecture

N. Schroer[f*], G. Crone[b], D. Della Volpe[c], B. Gorini[d], B. Green[a], M. Joos[d], G. Kieft[e], K. Kordas[h]
A. Kugel[f], A. Misiejuk[a], P. Teixeira-Dias[a], L. Tremblet[d], J. Vermeulen[e], F. Wickens[g], P. Werner[d]

[a] Royal Holloway University of London, [b] University College London, [c] Universita & INFN Napoli
[d] CERN, [e] Nikhef Amsterdam, [f] Ruprecht-Karls-Universitaet Heidelberg
[g] Rutherford Appleton Laboratory, [h] University Bern

Abstract

About 600 custom-built **R**ead**O**ut **B**uffer **IN**put (ROBIN) PCI boards are used in the DataCollection system of the ATLAS experiment at CERN. They are plugged into the PCI slots of about 150 PCs of the ReadOut System (ROS). In the standard *bus-based* setup of the ROS requests and event data are passed via the PCI interfaces. The performance meets the requirements, but may need to be enhanced for more demanding use cases. Modifications in the software and firmware of the ROBINs have made it possible to improve the performance by using the on-board Gigabit Ethernet interfaces for passing part of the requests and of the data in the so called *switch-based* scenario. Details of these modifications as well as measurement results are presented in this paper.

I. Introduction

The first level trigger (L1) of the ATLAS experiment [1] at CERN reduces the event rate from 40 MHz (bunch crossing frequency of the LHC) to at maximum 100 kHz. With this input frequency fragment data is written to the buffers of custom made circuit boards (ROBIN [2]) at about 120 GB/s (via ∼ 1600 optical links, 3 per ROBIN). Typically four ROBINs are plugged into the PCI slots of each of the 150 ReadOut System (ROS) PCs and the read out of the event data is performed on the PCI bus, thus the name *bus-based* for this setup. The connection to the Data Collection (DC) network, which manages the selection and storage of events for later analysis, uses two of the four Gigabit Ethernet ports of a quad-port NIC plugged into the ROS PC. Only two interfaces are used as the CPU of the PC needs to handle the network protocol and its performance cannot cope with more. This is the main bottleneck of the *bus-based* scenario. In the standard use case the second level trigger (L2) requests data from 2-3 of the 12 links of a ROS PC (in the typical case of 4 ROBINs) at about 20 kHz and based on the L2 trigger decision the event builder system requests data at ∼3 kHz from all links. For use cases with higher L2 request rates or for trigger types which have additional bandwidth demands such as Inner Detector or Calorimeter full scans this setup cannot deliver sufficient performance. The ROBINs have the potential to be directly connected to the DC network with their built-in GbE ports in the so called *switch-based* scenario, which also allows the message handling to be offloaded to the PowerPC (PPC) processor[3] on the ROBIN. For this so far unused approach the FPGA[4] and PPC code of the ROBIN needed to be modified in order to improve the performance of the network interface and to adapt the message handling to the demands of the DC network.

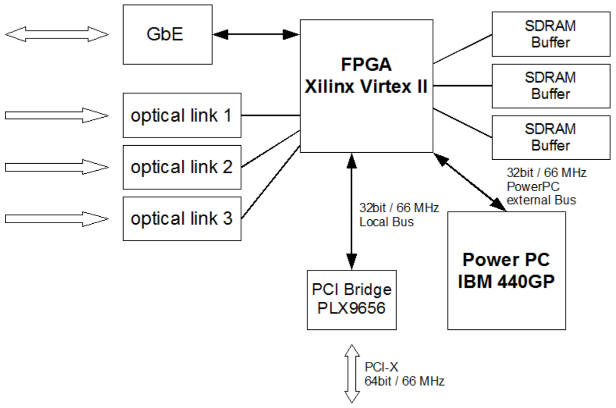

Figure 1: The main components and interfaces of the ROBIN.

A. *Modifications*

On a ROBIN the two main components are the FPGA and the PPC. Their original firmware is fully functional, but the built-in network interface is not optimized for the communication with the DC network. The throughput is limited to about $\frac{2}{3}$ (i.e. 80 MB/s) of the Gigabit Ethernet capacity and the maximal L1 rate is only around 60 kHz for the use case of 1kB fragments and a request ratio of 23%.
The firmware of the ROBIN has been modified to respond to messages at the network interface in the same way as a ROS PC to allow the integration into the DC network. However as most of the resources of the FPGA are already in use and the remaining ones are not sufficient to implement TCP, which is the standard protocol of the DC network, only UDP is supported. In the original firmware fragments with the same L1 ID (but from a different input channel) need to be requested individually and are sent out in one message per fragment. This has been improved to allow data from the three input links to be bundled in one message to minimize overhead in the transmissions by reducing the necessary requests. Due to the bundling the mes-

*Corresponding author, Email address: nschroer@cern.ch sages are bigger and reach the Ethernet frame limit of 1.5kB earlier and need to be divided into several Ethernet packets. To avoid this and to improve the performance support of jumbo Ethernet frames of up to 6kB is available in the new firmware. Furthermore the possibility to use DMA for internal data handling is now fully operational and allows buffering of incoming fragments in parallel to message processing and speeds up packet building by managing the transfer of header and data to the output buffer. As well as the modifications necessary to allow the support of jumbo frames, the latest FPGA firmware was improved by adding a second buffer to the transmission part of the network interface. This additional buffer allows the DMA engine to complete one packet while an already completed one can be transmitted, thus minimizing the send latency.

B. *Test Setup*

For the test setup the ROBIN is housed in a ROS PC and another PC is used to run a test program to simulate the DC network. This test program requests data fragments and sends delete messages (with 100 delete commands per message) to free the ROBIN buffers. UDP is used to communicate via a direct network connection between the ROBINs NIC and the requesting PC. Event fragments are generated by the internal data generator of the ROBIN, which has been implemented for test purposes. The size of the fragments can be programmed. The rate is throttled if the buffers of the ROBIN are full. This is the situation for the measurements described in this paper, therefore this rate is equal to the delete rate. The test program has reduced functionality compared to the ROS software environment that is usually used to request and delete the fragments via the bus interface, but it is easier to setup and suffices for performance measurements. The goal is to be able to request fragments of 1kB at a rate of 23 kHz from all three links while the input rate (L1 rate) is 100 kHz, which corresponds to requesting 23% of the data. As it is not possible in this test setup to set the L1 rate to a given value, the request frequency and associated throughput are measured for different fractions of the events (generated by the internal data generator) requested and for different fragment sizes. Hence the maximal possible L1 frequency is determined by multiplying the measured request rate by $\frac{100}{\%requested}$.

II. Measurement results

In figure 2 measurement results for the target request ratio of 23% are presented for both the original and the modified firmware. Contrary to the original firmware, which is not able to service a L1 rate of 100 kHz the modified firmware is capable of doing so up to a fragment size of about 1.25 kB (∼300 words of 4 bytes) which exceeds the requirement. The gain in manageable L1 rate is more than 75%.

To measure the maximal possible throughput of the network interface 100% of the fragments are requested which keeps the fraction of time spent on managing fragments (buffering & deleting) as small as possible. The results are shown in figure 3. The throughput of the modified firmware is increased by about 50% compared to the original firmware, reaching the limit of Gigabit Ethernet of ∼120 MB/s.

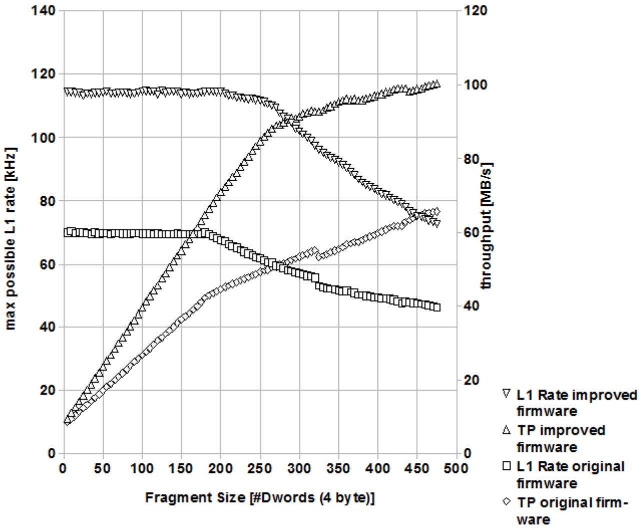

Figure 2: Results of measurements to determine the maximum L1 rate at target request ratio of 23% from all 3 links. Throughput and calculated (actual measured request rate * 100/23) maximal possible L1 rate as a function of fragment size. Both measurements are done in the switch-based setup, one with the original and the other with the modified firmware.

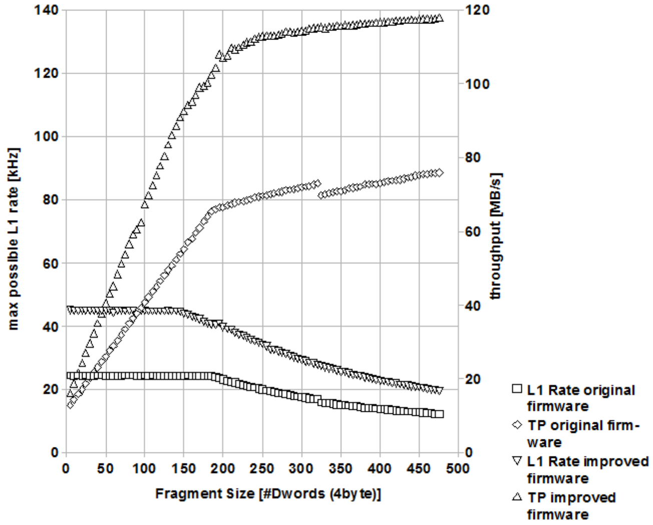

Figure 3: Results of measurements to determine maximum throughput by requesting 100% of fragment data from all 3 links.

The graphs of the measurement results show that for small fragment sizes the request rate (and thus the maximum L1 rate) does not depend on the fragment size and therefore the throughput is increasing linearly with bigger fragment sizes. Request handling and the deletion of fragments are overlapping with the data transfer which can be performed in parallel by the DMA.

As long as the data transfer time is shorter than the processing time of the requests the latter is dominating and thus results in constant event rate. For fragments larger than about 175 or 275 words, depending on the version of the firmware, the internal data transfers no longer overlap completely with processing by the processor. Therefore the event rate decreases for increasing fragment size.

Finally the request rate at a fixed L1 rate of 100 kHz is calculated from the data of the prior and several other measurements (see figure 4). These are the most significant figures as they represent the use case studied. As expected from the measurements with 23% request ratio the original firmware cannot provide a performance which would allow to request with 23 kHz. But with the modifications about three times the request rate is feasible fulfilling the requirements for fragment sizes of up to 1.25 kB.

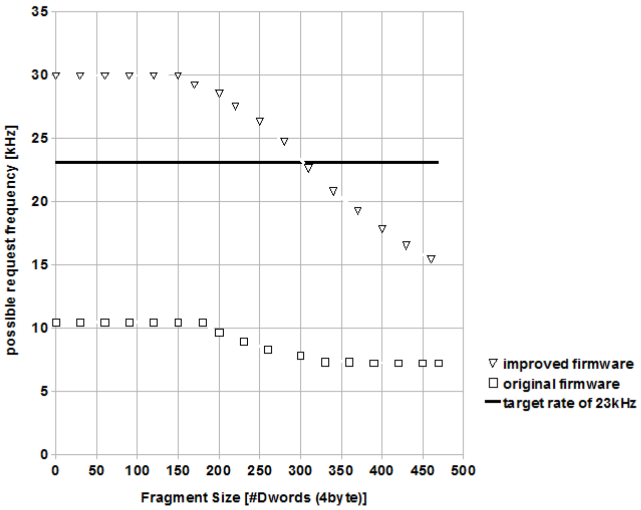

Figure 4: Calculated maximum request rate for a fixed L1 rate of 100 kHz.

III. Conclusions

The modifications of the ROBIN firmware result in a significant performance increase of the network interface, making it possible to request event data of up to 1.25kB (∼300 words of 4 byte) per fragment with more than 23 kHz at a fixed L1 rate of 100 kHz, hence fulfilling all the requirements. Used in switch-based mode each ROBIN can provide more than half of the output data rate of a ROS PC in the bus-based scenario, therefore the 4 ROBINs installed in a typical ROS PC together can provide over twice the output. This yields the potential to consider use cases with high L2 request rates or for trigger types which have additional bandwidth demands such as Inner Detector or Calorimeter full scans. With the modification of the message handling of the network interface to the standard format used in the DC network, an integration into the system is fairly straightforward, although additional cabling is required as each ROBIN needs to be connected to a switch. This setup would be used only in those parts of the readout system with high demands, thus the amount of extra cabling and switches is modest. Tradeoffs are that the switches need to be able to handle jumbo frames and that only UDP can be used to communicate directly with the ROBIN. But it is remarkable that the hardware design of our board together with reconfigurable components could be used to optimize the performance and implement alternative data transfer solutions.

References

[1] The ATLAS Collaboration, G. Aad et al., The ATLAS Experiment at the CERN Large Hadron Collider, JINST 3 (2008)S08003.

[2] R, Cranfield et al., The ATLAS ROBIN, JINST 3 (2008) T01002.

[3] IBM PowerPC 440GP embedded processor 462 MHz http://www.alacron.com/downloads/vncl98076xz/440GP_pb.pdf

[4] Xilinx Virtex II XC2V2000 FPGA http://www.xilinx.com/support/#Virtex-II