A multiple 10 Gbit Ethernet data transfer system for EIGER

M. Brückner, A. Bergamaschi, S. Cartier, R. Dinapoli, E. Fröjdh, D. Greiffenberg, I. Johnson, D. Mayilyan, D. Mezza, A. Mozzanica, M. Ramilli, S. Redford, C. Ruder, L. Schädler, E. Schmid, B. Schmitt, X. Shi, D. Thattil, G. Theidel, G. Tinti, J. Zhang

***Abstract*—Eiger is a single-photon counting x-ray pixel detector developed at the Paul Scherrer Institute for energies up to 25 keV with a pixel size of 75×75 µm². The Eiger detector is designed for synchroton applications and consists of several modules each having a total of 500 kpixels. 1.5 Mpixel and 2 Mpixel detectors (three and four modules) are being integrated at several beamlines and a 9 Mpixel detector (18 modules) is currently under construction.**

An Eiger module is subdivided into two half modules each having its independent but overall synchronized readout system consisting of a front-end board and a back-end board. The maximum frame rate is 22 kHz independent of the detector size. The data input stream is sorted in two FPGAs on the front-end board. The data rate here goes up to 22 Gbit/s. A rate correction is applied to compensate for the counting loss at high count rates. Then the data stream is sent over eight 3.125 Gbit/s highspeed transceivers to the back-end board which receives and further processes the data. On the back-end board the stream is buffered in a DDR2 memory. This allows image summation to extend the dynamic range from 12 bits to 32 bits and also extends the limited external data rate of the 10 Gbit/s Ethernet interface per half module for a short duration. On a 9 Mpixel detector, with its 36 half modules, the maximum data rate reaches 45 GByte/s. To reduce the network load on the servers side a round robin procedure is implemented by sending the stream to several servers. Here the challenge of keeping the images in one piece has been taken into account. A second approach is online compression currently being implemented to reduce the network load and to widen the Ethernet bottleneck. The firmware layout as well as the presented and implemented functions will be presented in detail.

I. Introduction

THE EIGER detector [1] is a single-photon counting x-ray pixel detector with a scalable size. In a module the silicon sensor with a size of approximately 8×4 cm² is 320 µm thick and bump-bonded to 4×2 readout chips. Each readout chip has 256×256 pixels with size of 75×75 µm² each. The analogue signal is pre-amplified and shaped inside the pixel cells in the readout chip and compared with the global threshold. Beside this there are local trimming bits for each cell correcting threshold dispersion. A count is added to a counter in the cell if the pulse height is above the threshold. To increase the readout speed it is possible to reduce the counter value from 12 bit to 8 bit or 4 bit. The dead time of the readout chip is only 4 µs because the counter value can be latched and readout while a new frame is taken. The corresponding frame rates are 8 kHz (12 bit), 12 kHz (8 bit) and 22 kHz (4 bit).

The readout chips are connected to either the top or the bottom half module of the readout system (4 chips to the top and 4 to the bottom). The half modules are equipped with identical hardware and firmware and receive from the readout chips up to 24 Gbit/s each.

On the half module image processing is done. Especially image summation which expands the dynamic range from 12 bit to 32 bit and rate correction which corrects the loss of linearity of the counter due to pile-up effects can be applied in firmware [2]. For the summation and buffering 4 GByte DDR2 memory is available.

Fig. 1. An EIGER 500k Module

The picture shown in Figure 1 is an 500k EIGER module. This is the basic element of larger systems like the 1.5 Mpixel, the 2 Mpixel or the 9 Mpixel detector. While the 9 Mpixel detector is currently under production the others are deployed to the coherent small-angle scattering (cSAXS) beamline at Paul Scherrer Institute (PSI) respectively the European Synchrotron Radiation Facility.

These multi-module detectors present the same readout characteristics (e.g. frame rate) as a half module due to the high parallelism of the detector's readout system.

Manuscript received May 30, 2016.

M. Brückner, A. Bergamaschi, S. Cartier, R. Dinapoli, E. Fröjdh, D. Greiffenberg, D. Mayilyan, D. Mezza, A. Mozzanica, M. Ramilli, S. Redford, C. Ruder, L. Schädler, E. Schmid, B. Schmitt, X. Shi, D. Thattil, G. Theidel, G. Tinti, J. Zhangare with the Paul Scherrer Institut, 5232 Villigen, Switzerland (telephone: +41-56-310-5439, e-mail: martin.brueckner@psi.ch).

I. Johnson is with the Engineering Division, Lawrence Berkeley Nation Laboratory, Berkeley, California, USA

978-1-5090-2014-0/16/$31.00 ©2016 IEEE

II. Hardware Overview

As described in the introduction an EIGER detector consists of several modules (3 modules in the 1.5 Mpixel detector and 18 modules in the 9 Mpixel detector) and the readout electronics of these modules is identical per half module. Therefore it is sufficient enough to describe only the EIGER half module readout system.

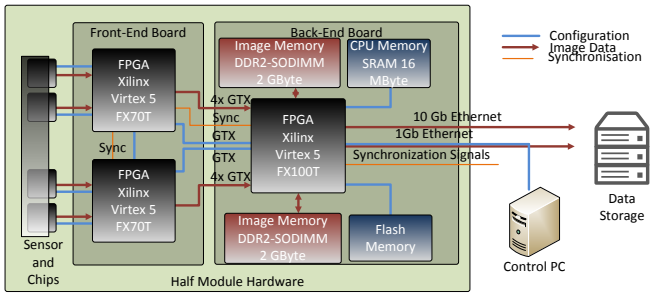

Fig. 2. EIGER half module with sensor, readout chips, front-end, and back-end board.

A half module consists of the sensor and its readout chips at the front followed by the front-end and back-end boards. In Figure 2 a schematic layout of the hardware is shown.

The front-end board has two Virtex 5 FX70T FPGAs (one FPGA for two readout chips). On the back end an Virtex 5 FX100T is used which is connected to two SO-DIMM DDR2 memory slots, to a 1 Gb/s Ethernet interface directly and a 10GbE AMCC-PHY. Linux runs on this FPGA using the integrated PowerPC440 CPU. For this purpose there is 32 MByte of SRAM on the board.

A ribbon cable connects all modules for synchronisation. One half module is software configured as master and the others as slaves. As described in Figure 3 a specific line is used by the slaves to indicate their state (pulling up means not ready).

III. Firmware

The readout system inside the FPGA is split up into tasks done in the front-end FPGAs and in the back-end FPGA. In Figure 4 the major firmware cores are shown.

While the front-end FPGAs mainly handle the DAQ state machine and the control of the readout chips, they do also reorder the incoming bits and apply a rate correction compensating the pile-up effect [2]. The loss in linearity of the counter occurs if a new photon hitting the sensor before the signal falls under the threshold. The correction is suitable for rates at least up to 1.2 MHz/pixel.

The maximum data rate is 24 Gbit/s independently if which counter depth is configured. However, the maximum frame rate drops with increasing bit depth. This is transferred over the GTX transceiver to the back-end board.

On the back-end board the data is stored in a DDR2 memory which acts like a big buffer since the 10 Gbit/s is not fast enough for the maximum data transfer speed. Additionally the memory is also used to increase the dynamic range from 12 bit on the ASIC to 32 bit by summing up incoming frames. This enhances the dynamic range. In this mode the DDR2

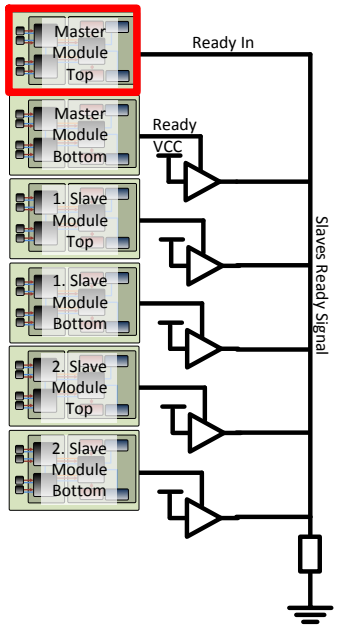

Fig. 3. Ready signal pulled up by the slaves when not ready and released otherwise. Method is proved to work for at least the 2 Mpixel detector with its 8 half modules.

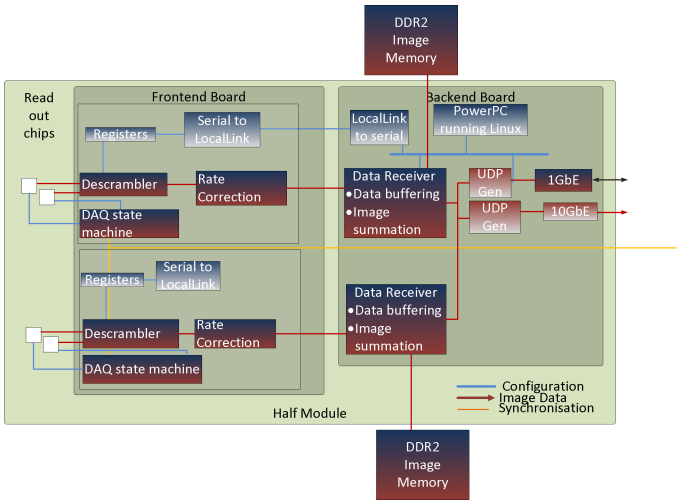

Fig. 4. Firmware Overview

memory which operates at 200 MHz has to handle three different data streams (incoming, outgoing for summation, and outgoing to Ethernet). Since the pixel bit-width is 32 bit here the theoretical maximum data rate is 6 GByte/s which exceeds the bandwidth of the memory. Therefore this 32 bit mode needs a reduced chip readout speed.

Figure 5 shows the maximum data rates at several positions of an EIGER half module depending of the counter bit width. Note that the 12 bit values are blown up to 16 bit in the front-end FPGA due to the rate correction. Therefore the maximum data rate is 32 GBit/s here.

The major bottleneck is still the Ethernet to the outer world. It limits the average data rate to 10 Gbit/s. Since data buffering is implemented the maximum data rate can be much higher for a limited number of frames. To widen that bottleneck a

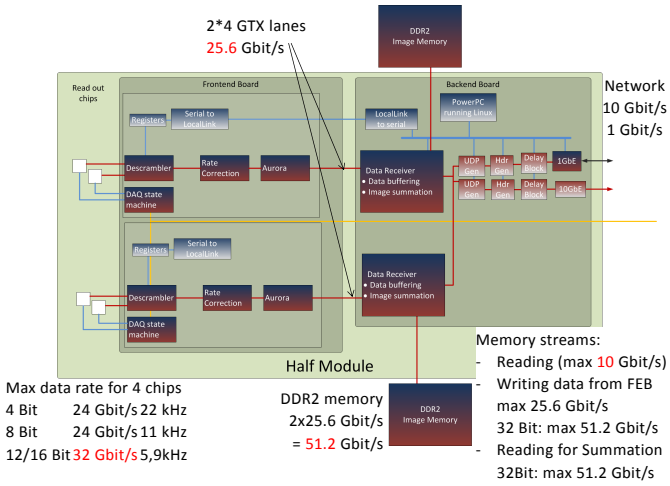

Fig. 5. Showing the maximum data rates at several points. It also shows the bottlenecks which are the memory in 32 bit mode and GTX transceivers in 16 bit mode as well as the 10 Gbit/s Ethernet.

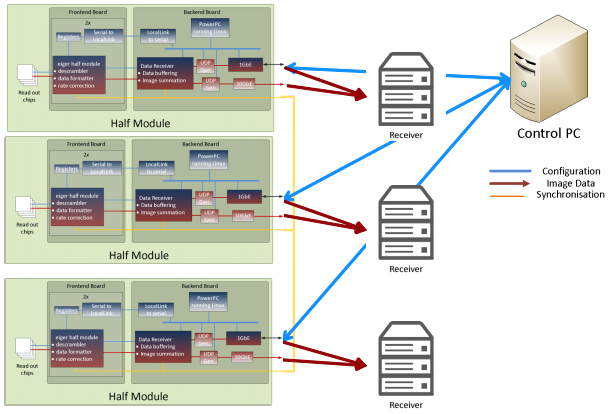

Fig. 6. Network configuration of the EIGER detector. Each half modules is connected to the control PC and to one receiver each.

compression is being implemented.

A. *Compression*

Typical frames made with EIGER or similar detectors have a lot of homogeneous areas. These areas can be compressed very well. A simple compression algorithm is being implemented. It subtracts neighbouring pixels and represents small differences with 4 bits only. Additionally bits indicating the width of the difference (4 bit or a 16 bit value) are stored in the header of each network packet. A 4096 Byte uncompressed packet has 2048 16 bit values. Storing the header information needs more 2048 bits in each packet (256 Byte). So in worst case the packet size increases to 4352 Byte but in best case it shrinks to 1280 Byte.

IV. Network Communication

There are two network connections for each half module: a control communication from a control PC and the UDP frame transmitting connection to the receiver. This is shown in Figure 6.

The control connection is a slow TCP connection from the PC to a self-written Linux software running in the PowerPC440 CPU inside the back-end board FPGA. The frame data from the front-end board bypasses the CPU and are sent directly to a UDP packet generator core. This core adds the corresponding UDP header to the data.

When using a 9 Mpixel detector which consists of 36 half modules the data rate is 360 Gb/s. Handling this on a single server is quite challenging. An easy option would be to distribute each frame to several servers so that the load per server is reduced. The disadvantage is that afterwards 36 frame fragments are distributed over several servers.

Therefore a delayed round robin is implemented. In the back-end board FPGA there is a block ram used which contains the UDP headers for several servers. An address counter increments when a frame has been sent so that the next frame gets another UDP header. Additionally each half module is configured with an individual send-delay counter value which is set up during initialization. This delays the sending of the frame packets. Assuming that the frame exposure time is less than the transmitting time it can happen that the first half module starts sending new frames while the others are still sending old data. In case of having only a few servers after a while it might happen that the first half module sending data to a server where old data are transmitted to in parallel. There is a second counter implemented preventing that a module starts transmitting new packets until a full frame was sent by all half modules. In Figure 7 the principle is shown. The advantage of this technique is that each frame remains in one piece or at least on the same server and and online visualization is possible.

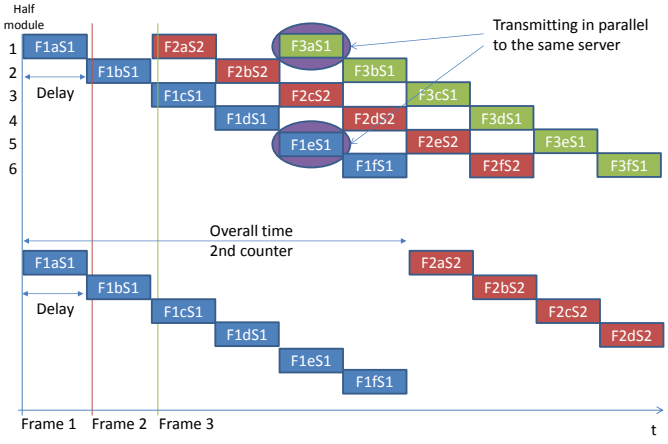

Fig. 7. Round Robin and Delay with two servers. In the upper part the possible conflict of sending to frames to a server at the same time (here *F3aS1* and *F1eS1*) is shown. (*F1aS1* means *Frame 1 Part a* send to *Server 1*)

V. Conclusion

The EIGER detector is a highly parallelized system which can be scaled from single to many several module systems. A 9 Mpixel detector is being built at the moment. Because of the independence of its half modules and frame rates up to 22 kHz the detector generates a big amount of data limited, however, by its 10 Gbit/s Ethernet connection per half module. To handle the data flow the data is sent to several servers using a round robin mechanism. To enhance the data flow further online compression is being implemented.

References

[1] R. Dinapoli et al., *EIGER: Next generation single photon counting detector for X-ray applications, Nucl. Instr. Meth. A*, **650** (2011) 79.

[2] I. Johnson et al., *Eiger: a single-photon counting x-ray detector, JINST*, **9** (2014) C05032.