ATL-DAQ-2004-020
26 November 2004

Design and Performance of a PCI Interface with four 2 Gbit/s Serial Optical Links

S. Haas, M. Joos

CERN, 1211 Geneva 23, Switzerland
Stefan.Haas@cern.ch, Markus.Joos@cern.ch

W. Iwanski

Henryk Niewodniczanski Institute of Nuclear Physics, Radzikowskiego 152, 31-342 Krakow, Poland
wieslaw.iwanski@ifj.edu.pl

Abstract

A reconfigurable PCI interface card (FILAR) with four on-board high-speed serial optical links has been developed for application in DAQ and test systems. FILAR cards, installed in low cost PCs, are currently being used in the combined test-beam of the ATLAS experiment at CERN as well as in several laboratory set-ups. The hardware and firmware design of the module and results from performance measurements are presented.

The four on-board 2 Gbit/s serial optical links conform to the S-LINK specification and are compatible with the Readout Link (ROL) implementation for the ATLAS experiment. The board design is largely based on FPGAs and the firmware uses a commercial 64-bit/66 MHz PCI IP core for the bus interface. Different firmware versions were developed which can be used to configure the hardware as either a data source or a destination card. Design optimizations have been made during the development cycle of the firmware to maximize the data throughput and reduce the PCI bus overhead as well as the CPU load. In a PC with multiple PCI segments an aggregate data throughput of over 1.5 Gbyte/s has been measured and transfer rates of more than 100 kHz have been achieved.

I. Introduction

Reliable high-speed data transfer is fundamental for the DAQ systems of current and future high-energy physics experiments. The S-LINK specification [1] developed at CERN is tailored for these type of applications. It defines the interfaces of the source and destination sides of a point-to-point data link with a bandwidth of 160 Mbyte/s (32 bits at 40 MHz) and features error detection, self-test functionality, flow control and return line signals.

In addition standard PCs are becoming increasingly popular for online data processing, because of their cost effectiveness and the ever increasing CPU performance. We present the design and performance of a reconfigurable PCI interface card (FILAR) with four on-board high-speed serial optical S-LINK channels.

First the FILAR PCI interface card hardware is introduced. The card is based on FPGAs and two different firmware versions were developed to implement transmitter and receiver functionality. The FPGA firmware and the necessary software to operate the card are then described. Design optimizations that were made to improve the performance and reduce the load on the host processor are also presented. Finally results from system performance measurements are shown.

II. FILAR Interface Card Hardware

The FILAR PCI interface card is designed to move data between four 2 Gbit/s S-LINK channels and a host processor with a 64-bit/66 MHz PCI bus. The board design is largely based on FPGAs which implement the S-LINK protocol, the PCI bus interface and the application specific logic. A commercial PCI IP core has been used for the bus interface [2]. Figure 1 below shows a picture of the FILAR card.

Figure 1: Picture of the FILAR PCI interface card

The FILAR card is based on the S32PCI64 design [3] which also featured a 64-bit PCI interface, but only had one slot for an S-LINK mezzanine card. By integrating four S-LINK channels onto the FILAR card, the PCI bus bandwidth of 528 Mbyte/sec can be used more efficiently. Given the limited number of PCI slots normally available on a PC motherboard it also allows to bring more channels into one computer. In addition it leads to a reduction of the cost per channel.

The four serial link channels on the FILAR card are compatible with the HOLA S-LINK implementation [4] which was developed at CERN for the Readout Link of the ATLAS experiment. About 1650 of these links will be used in ATLAS to transfer the data from the front-end electronics interface modules to the readout subsystem [5]. Each of the HOLA S-LINK interfaces consists of a standard pluggable fibre optic transceiver module [6], a serializer/deserializer chip and a small Altera APEX FPGA to implement the S-LINK protocol. The serial link speed is 2 Gbit/s, which allows for the overhead of the 8B10B encoding and the S-LINK protocol, while maintaining the required data throughput of 160 Mbyte/s.

The PCI interface logic is implemented in a larger FPGA (EP20K200C) from Altera. This device contains the PCI IP core as well as the application specific logic. Several firmware versions have been developed, which allow the card to be used either as a quad S-LINK receiver or transmitter. The firmware is described below.

III. S-LINK TO PCI RECEIVER FIRMWARE

The FILAR receiver firmware is designed to move data from four S-LINK channels via the PCI bus into the host computer's memory.

The data transfer is entirely handled by the interface card: the host processor only has to provide a list of physical addresses where the received data fragments are to be stored in memory. Once these addresses are written into a request FIFO buffer on the card, the firmware autonomously transfers the data into the host PC's memory using DMA. The status and the length of each data fragment received are then stored in an acknowledge FIFO buffer on the interface card, to be read out by the host CPU. The control words, which delimit the S-LINK data fragments, are also stored and can optionally be read out by the CPU as well. In addition the firmware compares the received control words with the ones defined by the ATLAS event format [7]. Figure 2 below shows a block diagram of the FILAR S-LINK to PCI receiver firmware.

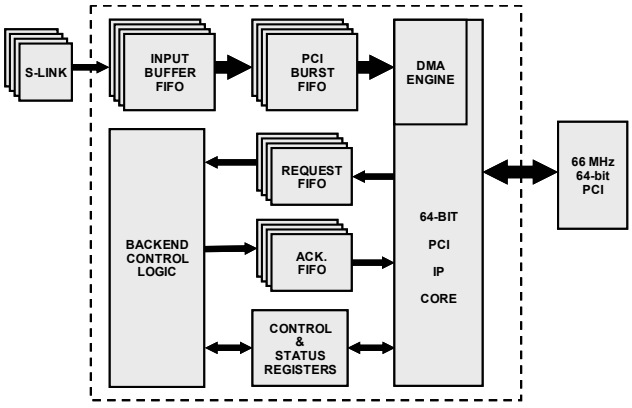

Figure 2: Receiver firmware block diagram

The memory for the data is allocated in fixed size pages; however the page size is configurable. If a data fragment is larger than the page size, then more than one request FIFO entry will be used. The DMA data transfer from the interface to the system memory uses PCI burst cycles to maximize the throughput. Data fragments of up to 1 Kbyte will be transferred in a single PCI write burst, longer data fragments will be split into several bursts by the interface.

In order to reduce the load on the host processor and the PCI protocol overheads we attempted to limit the number of PCI cycles required to operate the FILAR card. Only two PCI single-cycles accesses are required for each data fragment: one to write the address of a free memory page and one to read the length and status of the received data fragment. This firmware version is therefore subsequently referred to as the single-cycle (SC) receiver firmware. One additional access is needed per block of data fragments (typically 24) to check the occupancy of the on-board request and acknowledge FIFO buffers. Eventually the interface can also generate an interrupt to notify the CPU that the number of data fragments received by any channel has reached a predefined threshold. This functionality is required by the Linux device driver (see section V below).

Although the performance of the single-cycle protocol firmware was satisfactory (see section VI below), measurements with a PCI bus analyzer showed that the single-cycle accesses required for handshaking the interface were limiting the throughput of the card in particular for short data fragments. Therefore a second version of the receiver firmware package was developed which also uses DMA operations to transfer the memory addresses and the fragment length and status values between the on-board FIFO buffers and the host PC's memory. This firmware version is subsequently referred to as the DMA protocol receiver firmware. However due to the limited memory resources available in the interface FPGA used, the DMA protocol version of the receiver firmware only supports three of the four S-LINK input channels.

IV. PCI TO S-LINK TRANSMITTER FIRMWARE

In addition to the receiver firmware presented above, another firmware version was developed which allows the FILAR card to be used as a quad S-LINK transmitter (QUEST). The firmware is designed to fetch data from the host computers memory via the PCI bus and transmit it on up to four S-LINK output channels.

The QUEST firmware follows the same approach as the performance optimized version of the FILAR receiver firmware, i.e. the host CPU prepares a block of physical memory addresses where the data to be sent is stored and notifies the interface card. The pointers to the data are then fetched by the interface card using bus master DMA. The firmware autonomously reads the data fragments from the host PC's memory using PCI burst transfers and transmits them on the four S-LINK output channels. Control words for framing can be automatically added if required.

V. SOFTWARE

The software for the FILAR card consists of a Linux device driver and a user library which can handle multiple FILAR cards installed in one PC. Specific Linux kernel drivers have been developed for each of the three firmware versions described above. The main focus during the development of the software was on maximising the throughput and on the ability of the kernel driver to handshake the FILAR hardware without permanent attention from the application layer.

The description given below applies to the receiver operation; however the same principles apply to the software for the transmitter firmware.

The decoupling of the driver level from the application code is achieved by a block of shared memory that can be accessed from either side. This shared memory can be seen as an extension of the request and acknowledge FIFOs implemented in the FILAR firmware. A user program can write the PCI addresses of many data buffers to the shared memory with one function call. If the driver detects that the request FIFO of the FILAR is not full it automatically reads new PCI addresses from the shared memory area and copies them into the hardware FIFO on the card. In the same way the driver writes the information about newly received data fragments to a shared memory area from where it will be picked up by the application in an asynchronous manner. This architecture does not only allow the application to process data fragments in large groups but also reduces the amount of context switches between the user and the kernel level.

At the user level the hardware is seen as an array of input channels, e.g. with three FILAR cards installed in the PC the application code can address channels 0 to 11 and does not have to care about the mapping of channels onto cards. The programmer only has to know how the mapping between PCI device numbers and PCI slots is done for his motherboard.

By means of a dynamic file in the proc file system the activity of the driver and the status of the hardware can be monitored at any time without the need to run special applications. The driver also supports writing to the proc file in order to be able to dynamically enable or disable debug output from the driver to the Linux logging system.

The software described above is fully integrated into the ATLAS dataflow project. It relies on other libraries of that project, e.g. for the allocation of contiguous memory blocks or the formatting of error messages. Some of the test programs for the FILAR hardware use another driver from that project which allows these user processes to get direct memory mapped access to the registers of the FILAR card.

VI. PERFORMANCE MEASUREMENTS

A. Hardware Configuration

The system performance measurements were performed using a PC-based configuration as shown in Figure 3 below.

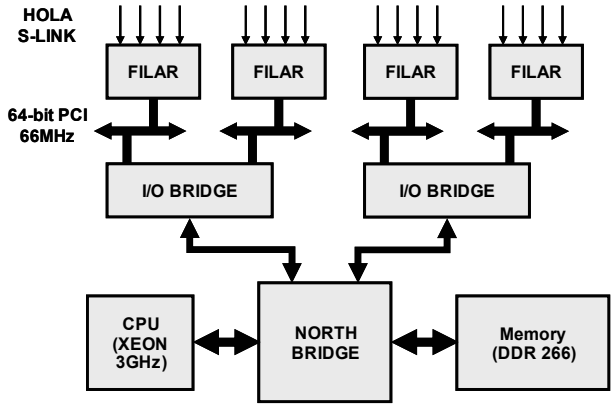

Figure 3: Hardware configuration

The processor (Intel Xeon) runs at 3 GHz and the motherboard (SuperMicro X5DL8-GG) features four independent 64-bit/66 MHz PCI bus segments. The architecture of the motherboard chipset is important for achieving a good system I/O performance, separate PCI bus segments each with a high-bandwidth connection to the main memory are essential.

The S-LINK input channels are connected to four FILAR interfaces, with only one card installed per PCI segment in order to maximise the input bandwidth. The S-LINK fragments were generated using FILAR cards programmed with a special data generator firmware capable of producing data with varying fragment sizes at the maximum link rate.

B. Receiver Performance

The performance of a single FILAR card has been measured and the results for the two different receiver firmware versions are compared. Figure 4 below shows the aggregate throughput of one FILAR card with one, two and three active input channels as a function of the fragment size for the single-cycle and the DMA protocol firmware.

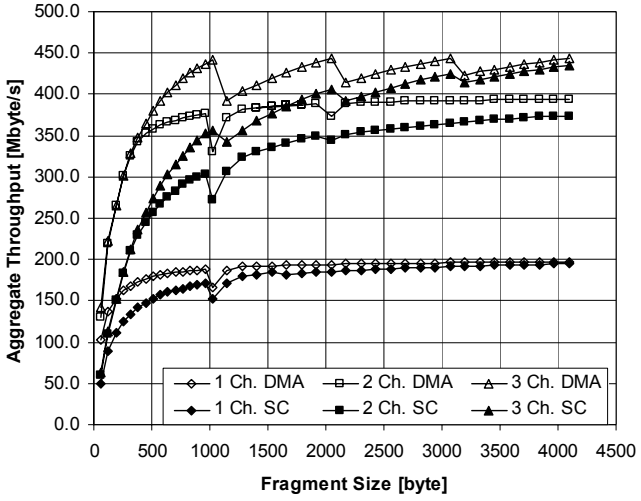

Figure 4: Receiver firmware performance

The results show a performance improvement of about 20% for the DMA-based protocol compared to the single-cycle firmware for data fragments of 1 Kbyte. For fragments of 4 Kbyte, the performance of both firmware versions is similar, since the protocol overhead is less significant for long data blocks.

For data fragment sizes above 1 Kbyte the measured performance scales with the number of input channels, however with three active inputs the throughput is limited by the available PCI bus bandwidth. The bus utilization measured with a PCI bus analyzer exceeds 90% in this case. The drop in throughput at intervals of 1 KByte is caused by the overhead for setting up a PCI burst. For data fragments of 4 Kbytes the measured throughput reaches about 450 Mbyte/s for the DMA protocol firmware. This allows for three input channels running at close to the target S-LINK speed of 160 Mbyte/s. For fragments of 500 bytes and less the performance is limited by the maximum receive rate, nevertheless the results demonstrate that a FILAR card can sustain an input event rate of over 140 kHz on three channels for data fragments of up to 1 Kbyte.

The performance of a system with up to four FILAR cards was also evaluated. The aggregate data throughput and the per-channel event rate were measured with 6, 9 and 12 active S-LINK input channels for various fragment sizes. The results are shown in Figure 5 below. The DMA protocol firmware was used for these tests.

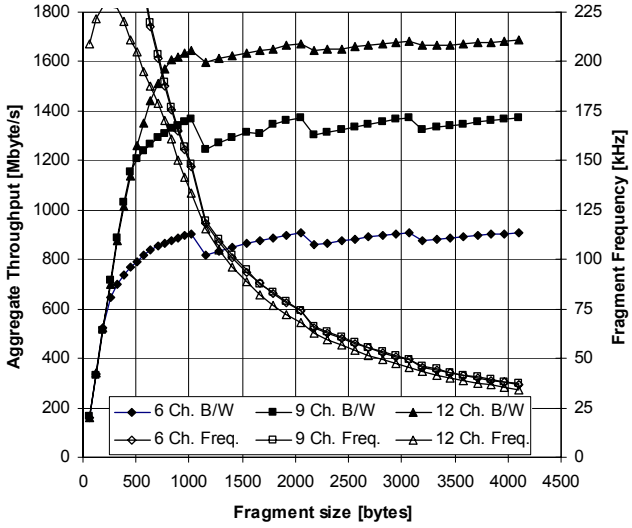

Figure 5: Performance of multiple FILAR cards

The results show that the throughput scales linearly for two and three FILAR cards and reaches an aggregate bandwidth of 1.7 Gbyte/s for four cards with a total of 12 active input channels. In this case the system performance is limited by the bandwidth of the host PC's memory. The setup can sustain an event rate of over 100 kHz for fragments of up to 1.3 Kbyte, even with 12 active inputs. It should however be noted that these results only demonstrate the maximum data transfer speed, and that no processing was done on the received data fragments.

C. Transmitter Performance

The performance of the S-LINK transmitter firmware was measured by connecting the transmitting card to a FILAR configured as a receiver installed in a second PC. Figure 6 below shows the measured throughput for the PCI to S-LINK interface with one to three active output channels as a function of the data fragment size.

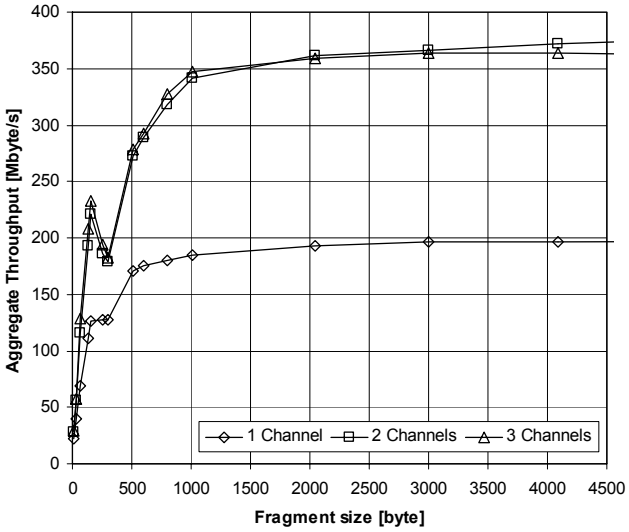

Figure 6: Transmitter firmware performance

The results show a maximum aggregate throughput of about 360 Mbyte/s, which is lower than the data rate achieved with the receiver firmware. The limitation is in the memory interface of the host PC, since read cycles from the memory are less efficient than write cycles. The available bandwidth is already saturated with 2 active output channels, enabling additional channels does not increase the aggregate throughput. The PCI to S-LINK interface can however sustain an event rate of over 100 kHz on three output links for data fragments of 1 Kbyte.

VII. SUMMARY

We have presented the hardware and firmware design of the FILAR PCI interface card. The card features four 2 Gbit/s serial optical S-LINK channels and a 64-bit/66 MHz PCI interface. Different firmware versions have been developed to configure the card either as a transmitter or a receiver. The firmware has been optimized for maximum data throughput while limiting the PCI bus overhead as well as the CPU load.

A software package consisting of a Linux device driver and a user library to manage the FILAR is also available. The package is integrated in the ATLAS dataflow software environment.

Measurements have shown an aggregate receive data rate of 1.7 GByte/s into memory in a PC with four FILAR cards. Event rates of over 100 kHz have been achieved for data fragments of up to 1 Kbyte on 12 input channels.

The FILAR card is being used in the combined ATLAS test beam as well as by different sub-detector groups for testing the readout of their of the various front-end electronics interface modules. Nearly 50 cards have been produced so far.

The performance of the current implementation is limited by the bandwidth of the 64-bit/66 MHz PCI interface. A redesign of the card with a 133 MHz PCI-X interface would double the bus bandwidth to over 1 Gbyte/s, thereby removing the bottleneck. Such an upgrade would be relatively straightforward since the PCI IP core used is also available for PCI-X. Newer technologies such as PCI-Express could provide even higher bandwidth. However the performance of the current implementation is sufficient for today's applications of the card and a redesign is therefore currently not planned.

VIII. ACKNOWLEDGEMENTS

The work presented was carried out within the framework of the ATLAS Trigger and Data Acquisition project and we gratefully acknowledge the contributions of other members of the TDAQ group.

This work was supported in part by the Polish State Committee for Research under Grant No. 620/E-77/SPUB-M/ CERN/P-03/DZ 110/2003-2005.

IX. REFERENCES

[1] O. Boyle, R. McLaren, E. van der Bij, "The S-LINK Interface Specification", CERN, March 1997. https://edms.cern.ch/file/110828/4/s-link.pdf

[2] PLD Applications PCI IP Core. http://www.plda.com

[3] W. Iwanski et al. "Evolution of S-LINK to PCI Interfaces", 8th Workshop on Electronics for LHC Experiments, Colmar, September 2002.

[4] HOLA High-speed Optical Link for ATLAS. http://hsi.web.cern.ch/HSI/s-link/devices/hola

[5] "The ATLAS HLT, DAQ & DCS Technical Design Report", CERN, October 2003.

[6] Small Form-factor Pluggable (SFP) Transceiver Multi-source Agreement (MSA). http://www.schelto.com/SFP/SFP%20MSA.pdf

[7] C. Bee et al. "The raw event format in the ATLAS Trigger & DAQ", CERN, February 2004. http://doc.cern.ch/archive/electronic/cern/others/atlnot/Note/daq/daq-98-129.pdf