The final design of the ATLAS Trigger/DAQ Readout-Buffer Input (ROBIN) Device

A. Kugel, R. Männer, M. Müller, M. Yu, E. Krause
University of Mannheim, B6, 23-29, 68131 Mannheim
mmueller@ti.uni-mannheim.de

B. Gorini, M. Joos, J. Petersen, S. Stancu
CERN, Geneva

B. Green, A. Misiejuk
Royal Holloway, University of London

G. Kieft
NIKHEF, Amsterdam

J. van Wasen
University of Mainz

Abstract

The ATLAS readout subsystem (ROS) is the main interface between 1600 detector front-end readout links (ROL) and the high level (HLT) trigger farms. Its core device, the readout-buffer input (ROBIN), accepts event data on 3 readout links (ROLs) with a maximum rate of 100 kHz and a bandwidth of up to 160 MB/s per link. Incoming event data is temporarily buffered and delivered via PCI or Gigabit Ethernet on request. Two devices, a XILINX XC2V2000 FPGA and an IBM PowerPC 440, are present, implementing the ROBIN's functionality. Furthermore one 64 MB SDRAM event data buffer is available per ROL. The device supports the ATLAS baseline implementation, which foresees the PCI bus as the main communication path inside the ROS, as well as an optional data path using Gigabit Ethernet to increase scalability when needed. The paper presents the final design of the ATLAS ROBIN. Measurement results, obtained with a prototype device in PCI bus and Gigabit Ethernet setups, show the usability and approve the design choices.

I. Introduction

The ATLAS readout subsystem (ROS) is one of the core devices of the ATLAS data acquisition (DAQ) chain. It acts as an event data buffer, while the level 2 trigger, a farm of 500 PCs, analyses the event. Event data, accepted by the level 1 trigger, arrive with a frequency of up to 75 kHz on 1600 readout links (ROL) from the ATLAS readout drivers (ROD). It is already planned to upgrade this rate to a value of 100 kHz later. Each ROL link delivers data with 160 MByte/s.

All event data, required by the level 2 processors for event selection, is delivered by the ROS via Gigabit Ethernet on demand. To reduce the amount of data, transferred between the ROS and the level 2 trigger processors, a detector region, called region-of-interest (RoI), is defined by the level 1 trigger. This RoI is a pair of angles, which describe a cone inside the ATLAS detector with its top in the experiment's interaction point. It restricts the area, and thus the event data fraction, which the level 2 processor has to evaluate to make the trigger decision. This requires the ROS to be able to deliver the event data arriving on each single ATLAS ROL separately to the level 2 processor on request. All event data accepted by the level 2 trigger leave the ROS via Gigabit Ethernet for event building. This is performed by the SFI event builder, a farm of approx. 50 PCs [1][2]. Events rejected by the level 2 processor get deleted from the ROS as soon as possible.

According to the ATLAS ROS baseline architecture the ROS is based on a commercial "of-the-shelf" high performance server PC with four PCI buses for a high I/O capacity. To handle the huge input volume, coming from the RODs, this PC is assisted by a number of custom hardware PCI boards called ROBIN (Readout Buffer Input). They process the ROL event data stream and buffer incoming data. A number of ROBIN prototypes have been evaluated in the past within ATLAS [3] [4]. This paper describes the hardware of the final ROBIN device. The device has been developed considering and evaluating all previous approaches. A first prototype stage of this final device has already been passed [5]. The results will be presented in this paper and prove the final ROBIN design choices.

II. ROS Requirements

Event data arrives at the ROS and thus at the ROBIN device with the level 1 accept rate. This is determined by the ATLAS trigger menu [6] to be approx. 25 kHz. Including a safety factor of three the complete ATLAS trigger and DAQ system is designed to sustain a level 1 rate of 75 kHz. This is planned to be upgraded to 100 kHz later. Since the ROBIN is a custom hardware device it cannot be upgraded so easily compared to PCs and network technology. Therefore it is the target of the ROBIN development

to support an input rate of 100 kHz already now.

Each of the 1600 RODs deliver one piece of the full event data called event fragment. Their size may vary between 200 Byte and 1200 Byte depending on the sub detector [2]. But the readout links are based on the HOLA SLink technology [7] with a nominal bandwidth of 160 MByte/s. Thus the ROBIN should be able to sustain this bandwidth too.

On the output site of the ROS all event data within a RoI, required for the level 2 decision, and all event data accepted by level 2 have to be delivered on request. This expected amount of data has been estimated by modelling [2]. For requests of RoI data the volume depends on the origin sub-detector. It is expected that data from the calorimeter sub-detector is required in most cases. Up to 7% of all data coming on one ROL from this sub-detector will be required by level 2 [2] in average. This leads to an expected event data fragment rate of 7 kHz per single ROL with a level 1 rate of 100 kHz for the mostly used sub-detector.

Additionally all events accepted by level 2 have to leave the ROS. Again modelling effort within the ATLAS community gives an estimation for the expected data volume. Up to 3% of all data arriving on one ROL will also be accepted by level 2. Thus, at a level 1 rate of 100 kHz, 3 kHz output rate has to be sustained per single ROL. Altogether up 10 kHz output rate per ROL have to be possible with the ATLAS ROS and the ROBIN implementation.

III. The ROS Baseline Architecture

Due to the difference between input and output load, described in the last section, a grouping of a number of ROLs to one output link is useful within a single ROS module. This is the basic approach of the ROS implementation. A ROS module is based on a standard, "off-the-shelf" server PC with four PCI buses for a high I/O capability. But to group in the order of 10 or 12 ROLs into one ROS module the PC must be "accelerated" by a custom input and buffer PCI boards, called ROBIN. These have the task to receive the event data and form the end-point of the ROLs. Furthermore the event data is buffered by the ROBINs and delivered via the PCI bus on demand. This baseline implementation of the ATLAS ROS is shown in Figure 1.

Incoming RoI requests from level 2, event data accepts, or deletes are initially processed by the ROS PC. They are distributed over the PCI bus to the ROBIN boards. Returning event data is collected and delivered to the requestor. The ROS module PC is the first instance of event building in case of an event accept decision by the level 2 trigger. All event data fragments are collected, merged and delivered to the SFI event builder farm. This partial local event building inside the ROS modules reduces the load, and with that the farm size, of the SFI.

Beside the PCI bus, which is the data path of the baseline implementation, a possible upgrade data path is already foreseen. In addition to the PCI bus it is intended to equip the ROBIN with an additional Gigabit Ethernet interface for a direct connection to the level 2 and event builder networks. This enables the possibility to send requests directly to the ROBIN device bypassing the ROS module PC. Event

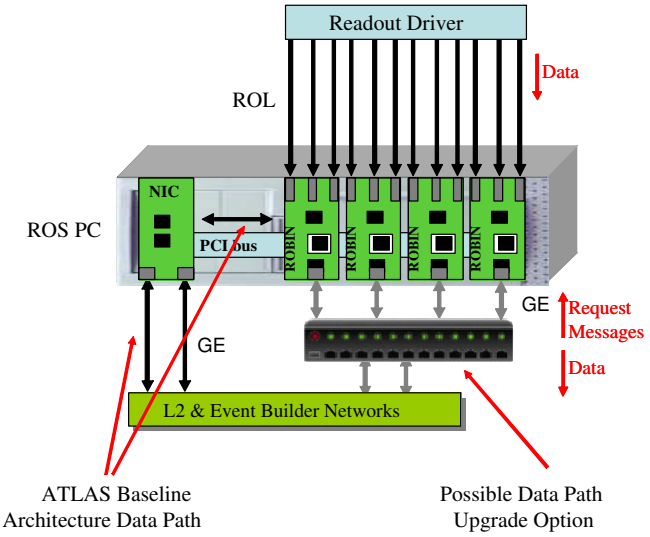

Figure 1: The ROS baseline architecture. A server PC forms the basis for a ROS module grouping a number of ROLs to one Gigabit Ethernet interface. The PCI bus is used as a main data path to communicate with the ROBIN boards. A direct Gigabit Ethernet connection to the ROBINs can be used as an upgrade option.

data delivery can also be performed through this interface. An additional Gigabit Ethernet switch, close to the ROS modules, concentrates the data streams from the ROBIN interfaces and provides the connectivity to the level 2 and event builder networks through two uplinks.

This additional data path can be used to improve the ROS output bandwidth in case of an unexpectedly high load (much above the estimated 10 kHz, see section II) between the ROS and the HLT.

IV. The final ATLAS ROBIN Device

A. Basic Considerations

The ROBIN device is a PCI board for receiving and buffering the event data fragments, which arrive over the ATLAS ROLs from the RODs. It is used together with a server PC to form the ATLAS ROS system according to the baseline architecture. On request from this ROS PC the ROBIN has to deliver or delete event data. Supplementary a Gigabit Ethernet interface should provide an additional data path as a later upgrade option. Requests to the ROBIN device are expected over both interfaces, the PCI bus and Gigabit Ethernet. Furthermore event data has to be delivered through both interfaces.

The ROBIN implementation must be able to fulfil the input and output requirements described in section II. To archive a high number of grouped ROLs in one PC, the ROBIN implementation should handle not only one ROL. It has been decided to implement three ROLs on one PCI ROBIN board. In this case all three ROL connectors and the Gigabit Ethernet interface must be accessible from outside the PC when the ROBIN is installed.

B. ROBIN Hardware Description

The hardware of the ROBIN device, shown in Figure 2, is based on two main parts: a Xilinx Virtex II 2000 FPGA in a FG896 package, and a PowerPC 440 microcontroller. Both are used to implement the ROBIN's main functionality. The connectivity to the ATLAS ROLs is provided by the three HOLA SLink connectors. They are implemented with 2 GBits/s optical transceivers and suitable serialisers / de-serialisers (TLK2501). The remaining parts, the SLink protocol engine, is embedded into the ROBIN FPGA as a VHDL core.

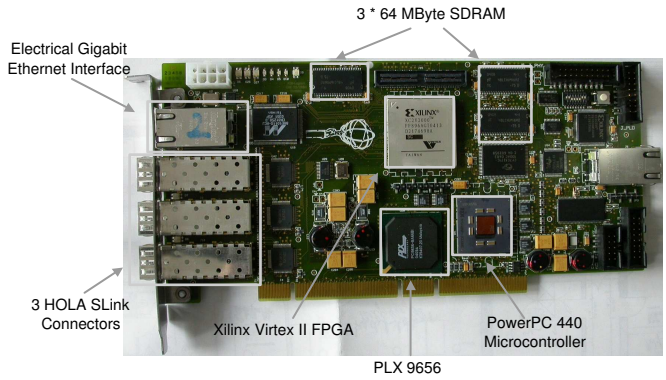

Figure 2: The ROBIN device. A FPGA and a PowerPC processor are available to implement the ROBIN functionality. The main connectivity is provided by three HOLA SLink connectors, a PCI bridge chip and an electrical Gigabit Ethernet interface.

Incoming event data is stored in a 64 MByte SDRAM buffer. For each of the three links one of these buffers is available. Both, PowerPC and FPGA have additional memory attached. The FPGA has 512 kByte ZBT SRAM to its disposal. It is intended to be used as a buffer to store for example incoming messages from Gigabit Ethernet. The PowerPC microcontroller has 128 MByte DDR SDRAM attached. It is used as general memory for the PowerPC application and to store a hash table for the event data buffer management.

The connection to the HLT farm (level 2 and SFI event builder) is provided by a PCI and an electrical Gigabit Ethernet interface. Event data or status and configuration information is exchanged over both interfaces on demand. The PCI interface is implemented with a PLX9656 intelligent bridge chip. It connects the ROBIN device to a 64 Bit / 66 MHz PCI bus with a nominal bandwidth of 528 MByte. On the ROBIN local side a 32 Bit / 66 MHz local bus with 264 MByte/s connects the PLX9656 to the FPGA. The Gigabit Ethernet interface is implemented by only an electrical connector and a Marvell 88E1011S PHY. The MAC layer is embedded into the FPGA as an IP core similar to the HOLA SLink. Therefore the Xilinx Gigabit Ethernet MAC core has been used.

ROBIN board control can be accomplished by the PowerPC or with limitations through the PCI bus and Gigabit Ethernet. This comprises firmware upload, control tasks such as setting the FPGA and PowerPC application parameters, and the collection of statistics. The firmware upload can be mainly done by the application running on the PowerPC through the FPGA's JTag interface. This is performed on the ROBIN's activation or can be initiated by a message via PCI bus or Gigabit Ethernet. The firmware data is stored in a flash memory device attached to the PowerPC and updateable via PCI and Gigabit Ethernet too. An alternative way to program the FPGA is provided through the PCI bus bypassing the PowerPC. This can be used in all cases when the PowerPC does not operate.

Several options for debugging have been added to the ROBIN device. A 50 pin header, directly connected to the FPGA, can be used to monitor any signal with a logic analyser. The PowerPC application provides a large number of debugging and status information covering the FPGA hardware and the PowerPC application itself. This is accessible through a simple serial interface with a standard Linux or Windows monitor application (e.g. Kermit). Additional debugging facilities are provided by the JTag interfaces of the FPGA, the PLX9656 or the PowerPC, which are accessible through a separate header. Finally a number of LEDs visualize the ROBIN device status.

C. ROBIN Hardware Usage

Figure 3 shows the various components of the ROBIN device, how they are connected, and the main data flow. The FPGA firmware, implemented in VHDL, processes all data, coming from the three ROLs, and stores it in one of the three event buffers. On request, either from PCI or Gigabit Ethernet, data is read from the buffers by the FPGA and transmitted over one of the links.

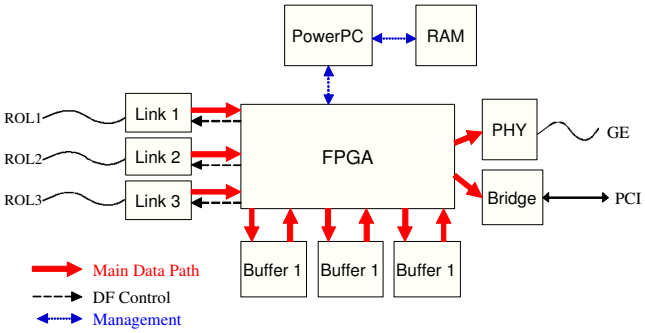

Figure 3: The hardware usage of the ROBIN application and main data paths. Incoming data travels only through the FPGA. The PowerPC has only management functionalities.

The PowerPC is never in the main data path. It performs only management and control functionalities implemented as a C program. This comprises the organisation of the event data buffer, decoding of incoming requests, and initiation of the reply of event data or status/debugging information.

C.1 Input and Buffer Management

The processing of incoming event data is shown in Figure 4. It is formatted according to the ATLAS Trigger and DAQ event format [8]. Data from the ROL is first stored in a 256 word FIFO. This decouples the HOLA SLink receiv-

ing procedure from the data processing within the ROBIN FPGA. An input handler reads the event data from this FIFO and extracts the event ID, assigned by the level 1 trigger, and the status information. The target buffer address to store the data is determined by the PowerPC application. It segments the buffer memory in pages of a pre-definable size, typically between 1 and 4 kByte. The address of this page is passed to the FPGA's input process by a "free-page-FIFO".

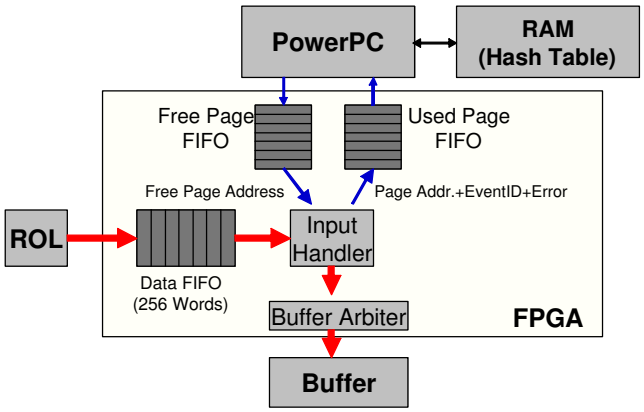

Figure 4: The handling of incoming event data from a ROL and the ROBIN buffer management.

When the event data has been stored on one or multiple buffer memory pages, the PowerPC microcontroller must be notified about the arrival of a new event fragment. Therefore the "used-page-FIFO" is present. It is filled by the FPGA with a value containing the event ID, the used page address, and the status information, which may contain SLink error transmission flags or errors detected by a rough data consistence check. This value is read by the PowerPC and used to build a hash table entry in the PowerPC's DDR SDRAM. This hash table entry can be used later to retrieve the position and status of the stored event fragment.

C.2 PCI Bus Messaging

To get event fragments or initiate a deletion of data on the ROBIN device, a request message must be sent. Also status and debugging information can be obtained and the ROBIN can be controlled via this communication channel. Figure 5 shows the message exchange procedure through the PCI bus for both directions: from the PC to the ROBIN and vice versa.

The PC is able to send messages with programmed I/O to the ROBIN. In this case the CPU of the PC has to write the message data word-by-word to a memory mapped region inside the ROBIN FPGA. Alternatively, for long messages (e.g. event delete messages with a whole bunch of event IDs for deletion), a PCI bus master DMA can be used for message transport. This instructs the PLX9656 to read the message data from the PCs memory and forward it to the FPGA. Both methods are used to write the message into a 2048 word dual-port RAM. Furthermore the address within this dual-port RAM has to be written

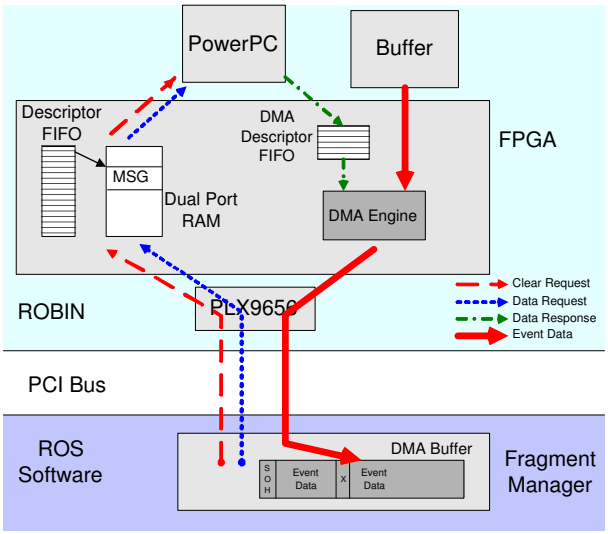

Figure 5: The PCI bus messaging scheme. Messages to the ROBIN, shown are request and delete messages, can be sent with programmed I/O or bus master DMA. The return channel uses only bus master DMA initiated by the ROBIN. The PC has to poll on the start-of-header marker (SOH), the first message word inside the DMA memory, to detect the arrival of messages from the ROBIN.

to a 32 word descriptor FIFO. Upon the entry in descriptor FIFO the PowerPC application reads the message, decodes and executes it. The message data contains a value encoding the requested service and the target address for replies. The complete format is described in [9].

Reply messages from the ROBIN hardware to the PC are initiated from the PowerPC microcontroller. These reply messages may contain event data or other information (status or debug information) supplied by the PowerPC. In any of these cases the PowerPC writes a three word command to the DMA descriptor FIFO, which is executed by the FPGA's DMA engine. This command specifies the amount of data to be transferred from an address inside the event data buffer to the PC DMA memory. Furthermore an arbitrary amount of data can be added by the PowerPC and is transmitted in advance. This allows the PowerPC to transport data to the PC (statistics or debug data). It is able to send event data messages plus an additional header required by the ATLAS event data format [8].

To signal the PC the arrival of data, a polling mechanism is supported by the ROBIN. Prior to a request message the PC application clears the first word of the target buffer at which it directs the reply message. Then the ROBIN transfers the reply data omitting the first word. The transfer is finalized with the transmission of the first word, which allows the PC software to detect the complete arrival of an event fragment or other information in the PC's memory.

C.3 Gigabit Ethernet Messaging

The processing of incoming messages via Gigabit Ethernet is similar to PCI (see Figure 6). Messages are read by the FPGA from the Ethernet MAC and stored in a dual-port RAM with the address placed in a descriptor FIFO. Currently this FIFO has only space for one word, which

allows only one message in the dual-port RAM. This is read and interpreted by the PowerPC. Reply messages are processed in the same way as with PCI except that it is not necessary to delay the transmission of the first word to indicate the finalisation of the transmission.

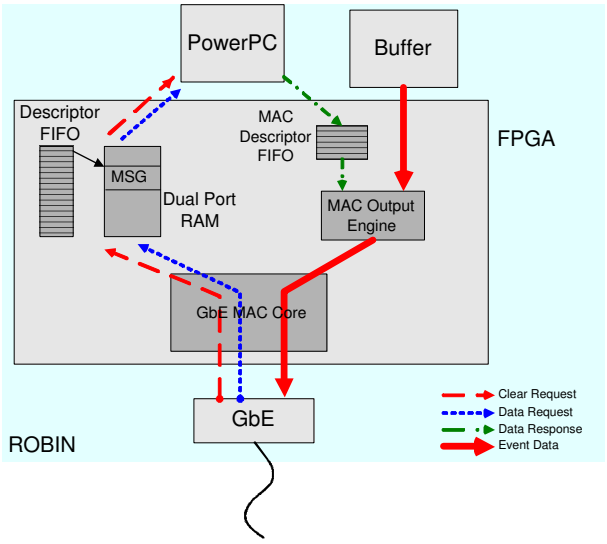

Figure 6: The messaging over Gigabit Ethernet. Event data request and delete messages are shown. They are temporary stored within the FPGA and read from the PowerPC. The reply messages are also initiated by the PowerPC and performed by the FPGA.

The exchange of messages via Gigabit Ethernet requires, in opposite to PCI, additional protocol headers and trailers. The current ROBIN implementation expects the standard Ethernet header and the raw-socket protocol [10]. The format of the message data contained in each Ethernet packet is equal to the messages sent over PCI.

V. ROBIN Test Status

To verify the design of the ROBIN board a prototype has been built prior to the final boards. This prototype provides only two ROLs and uses the slower PowerPC 405 microcontroller. Apart of that the design and the components are equal to the final device. The prototype has been tested in a PCI bus environment equal to the ATLAS baseline architecture and in a Gigabit Ethernet test setup.

A. Performance Measurements with PCI Bus setup

The PCI bus setup was based on a 2.4 GHz Xeon system with four PCI segments. Up to four ROBIN prototype boards (8 ROLs) have been used within this PC. The PC itself runs an application called "standaloneROS". This uses an internal trigger generator to simulate incoming event fragment and delete requests from the HLT farms. Upon such a generated request the ROBIN boards are accessed and returning event data is collected and merged. Without Gigabit Ethernet connection, the collected event data from the ROBINs have been processed until the point where they would have been sent to HLT and deleted then. Event data input to the ROBIN boards has been generated by a FPGA internal event data generator with an event rate of 130 kHz. This generates true input load equal to ROL SLink input.

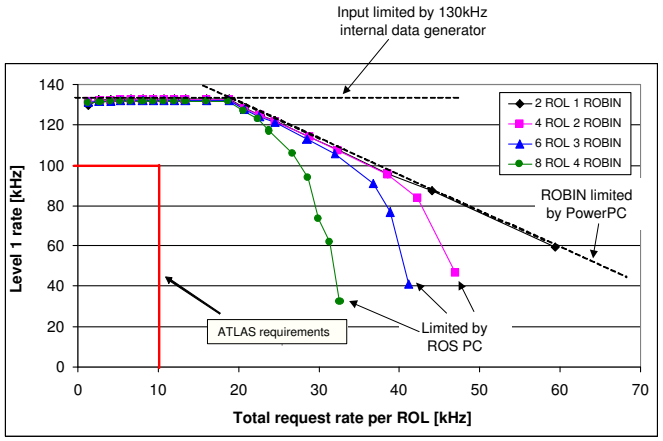

Figure 7: PCI performance test results for one to four ROBIN boards. The diagram shows the sustained level 1 accept rate versus the ROBIN request rate for 1 kByte fragments.

Figure 7 shows the measured performance of the system for one, two , three and four ROBIN boards. The y-axis displays the maximum sustained level 1 accept rate, which is the input rate to the ROBIN boards. The x-axis displays the possible request rate with which data can be pulled from the ROBINs. Each measurement point of a single curve has be obtained with a different fraction of incoming and requested event data fragments. All points at the right end of the curves show the request and level 1 rate when all incoming events are also requested over PCI. In this case the incoming level 1 rate is equal to the request rate. Moving to the left, each data point is measured with less event data requested. In this case more data get only deleted and never leave the ROBIN board. At the end no event data is requested any more. Only delete messages are sent to remove the incoming data from the buffer memory immediately.

For small request rates up to 20 kHz the level 1 (input) rate is equal to the internal data generator rate. For larger request rates the level 1 rate drops linearly independent of the number of ROBINs. This is caused by the limitations of the PowerPC. It cannot process the buffer management for incoming data fast enough, while requests from PCI have to be handled with a high rate. The maximum sustainable ROBIN input rate decreases with the increasing request rate. In the end the maximum PCI request rate is limited by the ability of the ROS PC application to handle the ROBIN boards. This depends on the number of boards in the system and explains the different maximum request rates for one to four boards. But in all cases the ATLAS requirements of 100 kHz input at 10 kHz request rate can be fulfilled by the ROBIN hardware.

B. Performance Measurements with Gigabit Ethernet setup

Two setups have been used to test the ROBIN hardware in a Gigabit Ethernet environment. The CERN setup im-

plemented a complete event builder slice comprising three PCs with the SFI event builder application, and one PC with the data flow manager application controlling the setup. All PCs were connected to one ROBIN prototype with a Gigabit Ethernet switch. The measurements with this setup have been carried out in an early stage of the prototype and thus with an old and un-optimized version of the FPGA firmware. The setup at the Royal Holloway University (RHUL) was simpler and used up to two requester PCs connected to the ROBIN, again with a Gigabit Ethernet switch.

In both setups the PCs continuously request event data from the ROBIN and delete it later. In case of the CERN setup the DFM instructs a SFI event builder PC to request all event fragments for a specific event from the ROBIN prototype and merge them to a single event data block. Later it sends an event delete message to the ROBIN. In case of the RHUL setup each of the two requestor PCs simply request and delete event data on the ROBIN continuously.

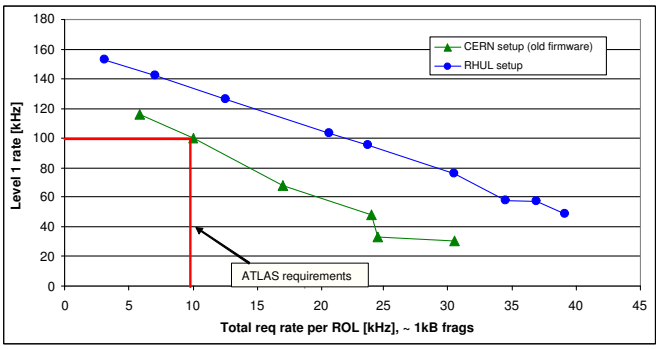

Figure 8: Gigabit Ethernet performance test results. The diagram shows the sustained level 1 accept rate versus the ROBIN request rate for 1 kByte fragments.

Figure 8 shows the resulting performance. Only one ROBIN board with two ROLs has been tested. The results are plotted just as in Figure 8. Again the ATLAS requirements can be fulfilled with both measurements.

VI. Conclusions

The decisions on the final implementation of the ATLAS ROS and its ROBIN boards has been done. The implementation is based on a Xeon server PC with multiple PCI buses. Furthermore a later upgrade option has already been foreseen. The ROBIN device supports the processing of event data from three ATLAS ROLs and is based on a FPGA assisted by a PowerPC microcontroller. Its design has been evaluated by two ROL prototypes in three different test setups. These have proven the ROBIN functionality and performance in a PCI bus and a Gigabit Ethernet environment. Another prove of the ROBIN design and firmware has been done recently in the ATLAS test beam setup. There a ROBIN prototype could be used successfully for event data processing in a realistic environment.

References

[1] J. C. Vermeulen et al, "The Baseline DataFlow System of the ATLAS Trigger & DAQ," in *9th Workshop on Electronics for LHC Experiments*, Amsterdam, Netherland, September 2003, pp. 147–151.

[2] ATLAS HLT/DAQ/DCS Group, *ATLAS High-Level Trigger, Data Acquisition and Controls Technical Design Report*, CERN, July 2003.

[3] ROS Community, "Robin summary," Tech. Rep., CERN, 2002, draft version: http://atlas.web.cern.ch/Atlas/GROUPS/DAQTRIG/ROS/documents/ROBINsummary.pdf.

[4] M. Müller, *Evaluation of an FPGA and PCI Bus based Readout Buffer for the Atlas Experiment*, Ph.D. thesis, Universität Mannheim, Aug. 2004.

[5] B. Gorini, M. Joos, J. Petersen, A. Kugel, R. Männer, M. Müller, M. Yu, B. Green, and G. Kieft, "A RobIn Prototype for a PCI-Bus based Atlas Readout-System," in *9th Workshop on Electronics for LHC Experiments*, Amsterdam, Netherland, September 2003, pp. 152–156.

[6] J. Vermeulen, V. Vercesi, and S. Tapprogge, "Beauty and the beast aka pesa and the ros," Presentation, ROS I/O Path Meeting, Dec. 2003.

[7] CERN, *HOLA High-speed Optical Link for Atlas*, http://hsi.web.cern.ch/HSI/s-link/devices/hola/datasheet.pdf.

[8] C. Bee, D. Francis, L. Mapelli, R. McLaren, G. Mornacchi, J. Petersen, and F. Wickens, "The event format in the atlas data acquisition," CERN ATLAS Note ATL-DAQ-98-129, CERN, Feb. 2004.

[9] B. Green, G.Kieft, and A. Kugel, "Atlas tdaq/dcs ros protoype-robin software interface," CERN EDMS Note ATL-DQ-EN-0003, CERN, Sept. 2002.

[10] S. Stancu, E. Kenzo, and M. LeVine, "Raw socket protocol description," CERN EDMS Note ATL-DQ-ES-0011, CERN, Apr. 2003.