Presented at IEEE Nuclear Science Symposium and Medical Imaging Conference, Lyon, France, 15-20 October 2000

A Read-Out Buffer Prototype for ATLAS High Level Triggers

D. Calvet, O. Gachelin, M. Huet, P. Le Dû, I. Mandjavidze, M. Mur

DAPNIA, CEA Saclay, 91191 Gif-sur-Yvette Cedex, France

Abstract

Read-Out Buffers are critical components in the dataflow chain of the ATLAS Trigger/DAQ system. At up to 75 kHz, after each Level-1 trigger accept signal, these devices receive and store digitized data from groups of front-end electronic channels. Several Read-Out Buffers are grouped to form a Read-Out Buffer Complex that acts as a data server for the High Level Triggers selection algorithms and for the final data collection system. This paper describes a functional prototype of a Read-Out Buffer based on a custom made PCI mezzanine card that is designed to accept input data at up to 160 MB/s, to store up to 8 MB of data and to distribute data chunks at the desired request rate. We describe the hardware of the card that is based on an Intel I960 processor and CPLDs. We present the integration of several of these cards in a Read-Out Buffer Complex. We measure various performance figures and we discuss to which extent these can fulfill ATLAS needs.

I. INTRODUCTION

With over 10^7 electronic channels and a bunch crossing rate of 40 MHz, the ATLAS experiment [1] will produce a massive amount of data. The detector read-out consists of ~1600 Read-Out Drivers (RODs). The Level-1 trigger identifies events at up to 75 kHz (upgradable to 100 kHz). Because the next level of trigger uses only a fraction of the event data and because the dataflow from the RODs is over 100 GB/s, an intermediate stage is needed between the RODs and the High Level Triggers (HLTs): the “Read-Out Buffers (ROBs)”. Their role is to absorb the dataflow from the RODs, provide HLT processors with the subset of data required to run the selection algorithms and buffer full event data as long as needed.

A detailed description of the requirements for the ROB and various prototype studies can be found in [2] and [3]. A ROB should be able to accept a sustained data flow of 135 MB/s (e.g. for calorimeter ROBs, blocks of 1.8 KB at 75 kHz) and provide data to HLTs at 1-10 MB/s. The buffer capacity should be sufficient to keep data during a 1-100 ms trigger latency. With a one-to-one ROD to ROB mapping, there will be ~1600 ROBs in the experiment.

The concept of our design is to have the input part and the event buffer of the ROB placed on a card called the Read-Out Buffer INput (ROBIN), and to share an output port between a small number of ROBINs. This forms a “ROB Complex”. We chose the popular PCI bus as interconnect between the ROBINs, within a ROB Complex.

II. DESIGN OF THE ROBIN

For flexibility during the prototyping phase, we designed the ROBIN on a PCI Mezzanine (PMC) form factor. A block diagram of the card is shown in Figure 1.

The card comprises a 96 MHz Intel I960 processor with an external program and data SRAM of 512 KB. The processor is in charge of local management and communication with the host via a PLX 9080 PCI bridge. External data is accepted by a 160 MB/s input port (40 MHz, 32 bit parallel) compliant with the S-link protocol [4]. This port is interfaced to the event memory (8 MB of SDRAM) via a 32 K-word input FIFO. The event memory can store up to 50 ms of event data arriving at 160 MB/s. There are two modes of operation for data generation. In normal mode, the input FIFO is filled with the data coming from the input port. For autonomous tests, an input stream can be generated internally by pre-loading the input FIFO via the host interface or the local processor. An external clock or an internal software trigger is used to control the rate of the simulated input. In both modes of operation, transfers from the input FIFO to the event memory take place at 64 MHz. For tests, it is also possible to re-program the logic of the input port to convert it to an output port. By connecting two ROBINs with a flat cable, one can act as an event data generator for the one being tested.

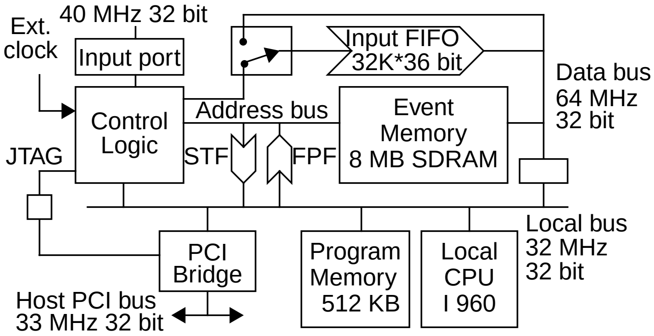


Figure 1: Functional diagram of the ROBIN PMC.

Programmable logic is used for the input port, to initiate transfers from the input FIFO to the event memory, to generate SDRAM addresses and refresh cycles... This logic consists in two 512 and one 256 macro-cells Complex Programmable Logic Devices (CPLDs). These components are programmed by the host processor using the serial line of the PCI bridge adapted to JTAG. Two 32 K-word FIFOs (noted STF and FPF) are also placed on the card. Their role is described in the next section. Components are placed on both sides of a 14-layer printed circuit board. Total power consumption is ~7.5 W.

III. PRINCIPLE OF OPERATION

To reach the target performance, a careful separation between the functions to be implemented in hardware and those that can be realized by software was made. On its input, the ROBIN has to cope with a very high rate of ~135 MB/s. Therefore the part that delineates data packets, performs elementary error checks and archives data blocks in the event memory has been entirely implemented with wired logic. Other functions, such as communicating with the host processor and locating in the event memory the blocks requested, are handled by the software running on the local CPU. No operating system is used at that level.

A. Data input section

The event memory is organized in pages of programmable size (256, 512 or 1 KB). At system initialization, the local processor fills the Free Page FIFO (FPF) with the address of each page. The input port is then ready to operate.

The control logic fetches the address of the first free page from the FPF and places a copy of this pointer in the STatus FIFO (STF). The input stream is transferred to the event memory via the Input FIFO. Particular words (e.g. Start/End of Event) are copied on the fly in the STF. When a page gets filled, subsequent free page addresses are fetched from the FPF and a copy of each page pointer is written in the STF. These operations proceed until an End of Event control word is received. Data transfers to the event memory stop, and a copy of the End of Event control word is placed in the STF. The control logic fetches a new free page address and resumes data transfers to the event memory.

Data blocks of different events cannot be interleaved on the input port, but the data of a given event can span across several non-consecutive pages in the event memory.

B. Local processor tasks

The local processor operates in parallel with the logic that handles the input stream. One of the first tasks of this processor is to unload the STF. When a complete event has been received without error, the words read from the STF are: 1) the address of the first page used for the event, 2) the Start of Event word, 3) the list of addresses of page used, 4) the End of Event Word. If these informations are available, the local processor reads the event identifier in the event memory and places it in a hash table maintained in the local SRAM. The entry in the hash table points to an event descriptor structure that contains the event identifier and the list of pointers to relevant event memory pages.

A second task of the local processor is the communication with the host processor via PCI. The host processor writes commands in the local processor memory via PCI; the local processor writes the replies in the host memory also via PCI. With such a scheme, both the local and host processors need not initiate a PCI transaction to see if a command or status was posted by its partner. Each actor simply accesses its respective local memory (polling). The host processor issues two main types of requests to the ROBIN: event data requests and event clear requests.

For an event data request, the host processor supplies the event identifier. The local processor searches for the corresponding entry in the event hash table. If no match is found, an error message is returned to the host processor. Hence the ROBIN cannot accept a request before the corresponding data has been received. If an event identifier match occurs, the local processor posts to the host processor a reply message with the list of event memory pages that contain the data of the requested event. No actual data movement takes place at this stage. The transfer of data from the ROBIN event memory to a PCI target (e.g. a network interface card or the host memory) has to be initiated by the host processor.

When the data of some events are no longer needed, requests to clear events are posted to the ROBIN. Although this is programmable, a clear message typically contains a list of 50 event identifiers. The local processor matches each event identifier to hash table entries, writes in the Free Page FIFO the list of page pointers that can now be re-used by the data input logic, clears the event descriptor structures, and remove the event identifier entries from the hash table.

The local processor also accumulates statistics and errors (e.g. number of event received, number of requests received and serviced, hash table miss, input link errors...) and delivers them to the host processor upon request.

IV. The ROB Complex

Several ROBINs can be attached to the same PCI bus and are controlled by a common host. This forms the so-called “ROB Complex” depicted in Figure 2. The network interface of the host (usually Ethernet) can be used for configuration and slow control while an additional network interface card (e.g. ATM) is used for fast communications and data transfers with the High Level Trigger processors and system.

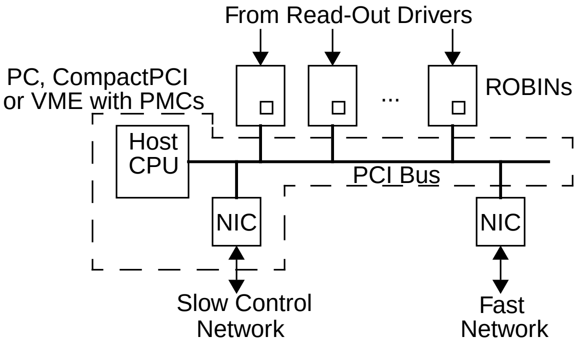


Figure 2: The ROB Complex.

There are several ways to build a ROB Complex. A first option uses the mechanics of a standard PC, with PMC to PCI passive adaptors. Another possible deployment is based on a VME single board computer. With PMC site extension cards, up to 5 ROBINs and one network interface card can be attached to the same host. The last option is to house the ROBINs in a CompactPCI chassis and use a host processor in CompactPCI format. In principle, this allows to build larger ROB Complexes (up to 8 slots in a PCI backplane without a PCI bridge, more if PCI bridges are used).

V. Performance Measurements

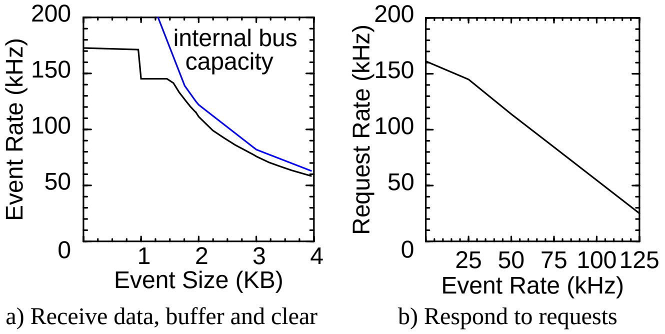
Because the ROB Complex is composed of many blocks interacting with each other, it is important to characterize each component independently to identify the limiting elements and better understand global performance. Various configurations have been assembled in VME/PMC, CompactPCI and standard PC environments. These were tested in stand-alone mode and integrated in an ATM demonstrator testbed. Some results have been reported in [5]. Because our VME and CompactPCI single board computers are not the most recent models, the performance tests presented in this paper were obtained on a 733 MHz Pentium III PC running Linux. This PC is equipped with a 155 Mbit/s ATM network interface and up to 4 ROBINs.

A. Characterization of the ROBIN

These measurements aim to characterize the performance of the various functions of a single ROBIN. Initially, minimum operation is made by the ROBIN. Then each function is turned on one after the other to identify at each step its impact on performance.

In a first test, we only check the operation of the input port and logic. One ROBIN acts as a generator. It is connected to the ROBIN being tested by a flat cable. The port logic operates correctly at up to 45 MHz, i.e. a data transfer rate of 180 MB/s. For data blocks of 1.8 KB, the maximum sustained event rate is 91 kHz. This satisfies the baseline requirement (75 kHz) but faster logic would be needed for the upgrade to run at 100 kHz.

A second test is to measure the capability of a ROBIN to accept events on its input, archive them in the event memory and free the event memory later on. In this configuration, input data is generated by the ROBIN internally; the size of events is varied. No requests for data are issued, the host processor simply posts requests to clear events to the ROBIN. We show in Figure 3.a the maximum rate that can be sustained for receiving, buffering and erasing events.



a) Receive data, buffer and clear

b) Respond to requests

Figure 3: Performance of the ROBIN.

For events that fit in one page of the event memory (1 KB), an event rate of 170 kHz can be sustained. This rate is determined by the execution time of the software running in the local processor. A step is observed when events need two pages to fit in the event memory. For events larger than 1.5 KB, the maximum rate of operation is mainly determined by the bandwidth of the internal bus between the input FIFO and the event memory (32 bit, 64 MHz). For events of 4 KB, the limit imposed by this bus is 64 kHz, the measurement is 60 kHz. For events of 1.8 KB, operation at 125 kHz is possible. These figures are upper limits: in a real application, requests for data are also present and a fraction of the internal bus bandwidth should be available for transferring data from the event memory to an external agent.

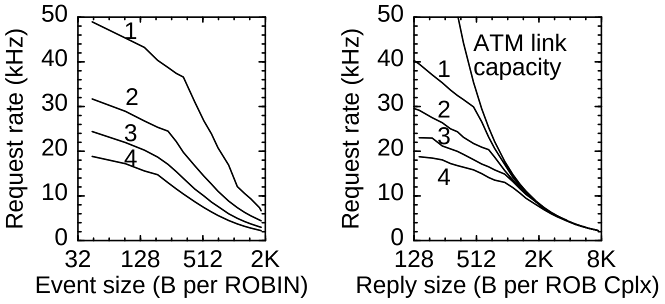
The second test is performed in the following conditions. The ROBIN generates events of 1.8 KB internally. These are received, stored and cleared at a rate that is varied. Data request generation is enabled in the host processor. For each event requested, the ROBIN locates the data in its event memory and posts the list of event memory pages concerned to the host processor. However, there is no transfer of the actual event data from the event memory of the ROBIN to an external device. We show in Figure 3.b the maximum rate of requests that can be serviced by the ROBIN when the event rate is varied. For low event rates, requests can be handled at 160 kHz (in this case the same event is requested several times). At the typical event rate of 75 kHz, requests can be handled at 80 kHz. In a real use of the ROBIN, the rate that can be effectively serviced will be lower because data need to be transferred outside of the event memory. The amount of time needed for this operation depends on the capability of the device that makes the transfer and the availability of the event memory data bus (that bus is also used for the transfers from the input FIFO to the event memory).

B. ROB Complex with an ATM connection

In this configuration, the data request and clear event messages are received by the host processor via ATM and posted to the ROBINs that are concerned. For data requests, when all ROBINs have replied to the host processor, the corresponding data are fetched from the event memory of each ROBIN by the DMA engine of the ATM network interface card, and are sent over the wire to the requester.

In a first test, the event input rate and the clear event rate are set to 75 kHz. The size of the event data generated to emulate the input is varied. The maximum rate of data requests that can be serviced is measured for a ROB Complex with 1, 2, 3 and 4 ROBINs. For each request, all the data for that event is sent to the requester. This mode of operation corresponds to event building / data acquisition where complete event data is needed. Measurements are shown in Figure 4.a. For short events, a ROB Complex with one ROBIN can service data requests at ~50 kHz. With two ROBINs, the rate achieved is more than half of the previous figure because some operations are performed in parallel. For large event sizes, the service rate is limited by the bandwidth of the ATM link. For events of 2 KB per ROBIN, a ROB Complex with 4 ROBINs can service requests for full event data at 2.1 kHz (i.e. a throughput of 16.8 MB/s, that is the saturation point of a 155 Mbit/s ATM link).

A second series of tests is performed at the nominal point of operation for ATLAS electro-magnetic calorimeter ROBs: the event input and event clear request rates are 75 kHz, data block size is 1.8 KB per event per ROBIN. The ROB Complex contains 4 ROBINs. Data requests concern 1, 2 or 3 ROBINs randomly selected among the 4, or all of them. For each request, only a fraction of the event data is returned. That amount of data is a parameter; its maximum value is 1.8 KB times the number of ROBINs concerned. This mode of operation corresponds to typical requests of ATLAS second level trigger, where only a part of the event data within a subset of ROBINs (spanning across several ROB Complexes) is of interest for the selection process.



a) Reply with all event data

b) Reply with selected data

Figure 4: Performance of the ROB Complex.

We show in Figure 4.b the maximum rate of operation that can be sustained. When only a small fraction of the event data is returned to the requester, the rate is determined by various software overheads, while for larger blocks, it is determined by the bandwidth of the output link.

In another test, we try to be as close as possible of a realistic point of operation for the final system. We verify that a ROB complex with 4 ROBINs is able to perform simultaneously the following tasks:

- receive and buffer events of 1.8 KB at 75 kHz,
- service requests for full event data at 1 kHz,
- service requests at 8 kHz where 1 ROBIN among 4 is concerned and half of the event data from that ROBIN is returned,
- service requests to clear events at 75 kHz.

These results are largely compatible with our current understanding of the requirements for ATLAS and show that the concepts of the ROBIN and the ROB Complex are valid.

VI. Pending Items and Discussion

Although satisfactory results have been obtained, some points still need further study.

At present, the network interface card attached to the ROB Complex has a 155 Mbit/s ATM link. When large blocks are sent, this link becomes the limiting point. Although the requirement is probably below that saturation point for many of the ROB Complexes in ATLAS, the most sollicitated ones could hit that limit. Tests could be made on a ROB Complex with a faster link (e.g. Gigabit Ethernet) to show a probable increase of the throughput on the output side of the system.

Because our ROBIN is a prototype, we chose the most flexible form factor, i.e. PMC. However, with this format, board space is scarce, available power is very limited and the input port had to be placed on the front panel. For the final system, robust mechanics is needed, and placing the input on the rear side of the card might be preferred. Our design could be re-deployed on a standard 3U Compact PCI card. In addition to robustness, this would also provide more board space and available power. Faster logic, a more powerful CPU and a larger event memory could easily be integrated. We are confident that with these possible improvements, the requirements for ATLAS would be more comfortably met.

VII. Summary and Conclusions

The design of a Read-Out Buffer INput PCI Mezzanine Card (ROBIN PMC) has been reported. It is based on programmable logic and includes an I960 processor. The card is capable of accepting an input stream of up to 160 MB/s and provides 8 MB of event buffer storage. The operation of the ROBIN has been detailed and its performance has been measured. A ROB Complex with up to 4 ROBINs and a 155 Mbit/s ATM network interface card has been assembled and tested in a standard 733 MHz desktop PC. Following a number of hypotheses, the performance achieved is within the range required for ATLAS. For example, events of 1.8 KB can be received, buffered during the required period and erased at 75 kHz, and simultaneously data corresponding to requested events can be retrieved from temporary storage and sent to the requesters at ATM wire speed.

VIII. References

[1] The ATLAS Collaboration, *ATLAS Technical Proposal, CERN LHCC 94/43*, 15 December 1994.

[2] R. Cranfield et al., “Options for the ROB Complex”, *ATLAS DAQ Note 2000-027*, 3 April 2000.

[3] The ATLAS Collaboration, *ATLAS High Level Triggers, DAQ and DCS Technical Proposal, CERN LHCC 2000/17*, 31 March 2000.

[4] H. C. van der Bij et al., “S-LINK, a data link interface specification for the LHC era”, *IEEE IXth Conference on Real Time Systems, Beaune, France*, 22-26 September 1997, pp. 199-203.

[5] J. Bystricky et al, “An integrated system for the ATLAS High Level Triggers: Concept, General Conclusions on Architecture Studies, Final Results of Prototyping with ATM”, *ATLAS DAQ Note 2000-011*, 10 March 2000.