View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by CERN Document Server

The COTS Approach to the Read-Out Crate in ATLAS DAQ Prototype -1

G. Ambrosini[b], H-P. Beck[a], S. Cetin[d], T. Conka[d], A. Fernandes[f], D. Francis[b], F. Hogbe Nlend[b], M. Joos[b], G. Lehmann[a], A. Mailov[c], L. Mapelli[b], G. Mornacchi[b], M. Niculescu[b,c], J. Petersen[b], D. Prigent[b], B. Rensch[e], L. Ribeiro Monteiro[f], J. Rochez[b], R. Spiwoks[b], L. Tremblet[b], G. Unel[b]

a. Laboratory for High Energy Physics, University of Bern, Switzerland.
b. CERN, Geneva, Switzerland.
c. Institute of Atomic Physics, Bucharest, Romania.
d. Bogazici University, Istanbul, Turkey.
e. Niels Bohr Institute, Copenhagen.
f. CERN, Geneva, Switzerland and Portugal

Abstract

A prototyping project has been undertaken by the ATLAS DAQ and Event Filter group. The aim is to design and implement a fully functional vertical slice of the ATLAS DAQ and Event Filter with maximum use of Commercial Off The Shelf components (COTS).

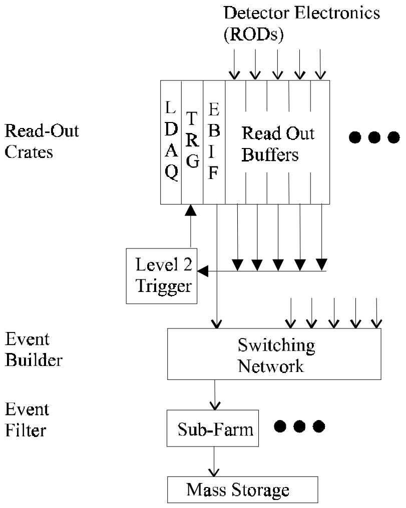
The Read-Out Crate is a modular component within the vertical slice whose principle functionality is to receive, buffer and forward detector data to the Event Filter systems via an event building network and to the Level 2 Trigger. As required by the project, the initial implementation is based on commercial components, namely VMEbus, PowerPC based single board computers and the Lynx-OS real-time operating system. The measured performance is compared to the results of a discrete event simulation of the Read-Out Crate using the PTOLEMY modelling tool. It has allowed us to model and study the Read-Out Crate performance based on a mixture of existing and forthcoming technologies, an example of the latter being VMEbus 2eSST, and different architectures. Results from studies in this area are also presented. The design and initial implementation of the Read-Out Crate is presented.

1. INTRODUCTION

The final design of the Data Acquisition (DAQ) and Event Filter (EF) system [1] for the ATLAS experiment at the LHC is not scheduled to start before 2000. However, due to the complexity of the system and to the severe requirements in terms of data rate and volume, hardware and software technologies must be evaluated and aspects of system integration studied before a final design can be implemented. The ATLAS/DAQ group has chosen to approach these investigations by building a fully functional prototype of the DAQ system [2] consisting of a complete "vertical slice" of the ATLAS DAQ/EF architecture including all the elements of an on-line system from detector read-out to data recording. Since it is understood that this prototype will not fulfil the final performance requirements it has been given the name DAQ/EF prototype "-1".

The DAQ/EF prototype –1 architecture contains a component which is responsible for receiving and buffering event fragments, event building and mass storage. This logical component, called the DataFlow is shown schematically in Figure 1.

Figure 1:DAQ/EF DataFlow architecture



The Read-Out Crates (ROCs) are responsible for moving the data between a subdetector and the Event Builder. Each of the Read-Out Buffers (ROBs) in a ROC receives and buffers detector event fragments of size ~1 KByte at a rate of ~100 kHz. This rate is determined by a Level 1 trigger system. To cope with the total amount of data from the detector (~100 Gbyte/s) a large number (~150) of ROCs is required.

A Level 2 Trigger system accesses a subset of the data in the ROBs - the so-called Regions Of Interest (RoI) - to provide an event rejection factor of ~100. For events accepted by the Level 2 Trigger, the associated event fragments within a crate are collected and sent via the Event Builder [3] to the Event Filter [4] where a final event selection based on the reconstruction of complete events is performed before the events are written to mass storage.

2. THE READ-OUT CRATE

A logical view of the ROC is shown in Figure 2 . It consists of a Local DAQ (LDAQ), one or more Read-Out Buffers, an Event Builder Interface (EBIF) and a Trigger (TRG). This logical view also foresees the possibility to collapse several logical modules (e.g. TRG and ROB) into one physical module. The ROB, EBIF and TRG are instances of a logical object referred to as an I/O Module (IOM)[6]. The different modules have the following functionality, related to the main data flow:

- The LDAQ provides the run control and monitoring functions within the ROC and communicates for that purpose with the other modules in the ROC.
- The TRG is responsible for the control of the data flow in the ROC. It receives and buffers data control messages from the trigger system and distributes them to other IOMs within the ROC. The types of data control messages are described below.
- The EBIF receives data control messages from the TRG. It collects and buffers event fragments from the ROBs via a process called Data Collection (DC). The crate fragments are output to the event builder.
- The functionality of the ROB includes the reception and buffering of event fragments from the detector Read-Out Drivers (RODs). In addition the ROB receives data control messages from the TRG and the EBIF which define the actions to be performed on the buffered event fragments. The ROB provides data to the EBIF for data collection and to the Level 2 Trigger system.

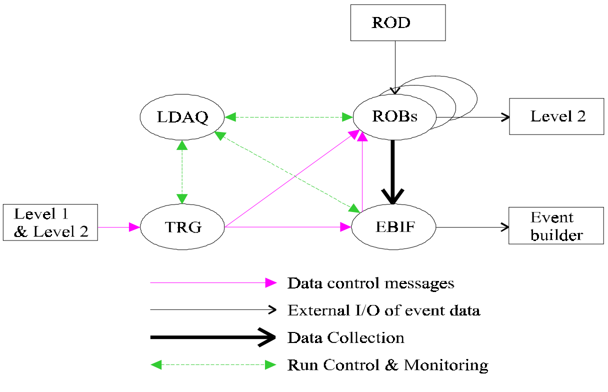


Figure 2: Logical model of the ROC

Data control messages are exchanged between the IOMs via an intra-crate message passing system. The following data control messages are defined:

- *Level 2 Reject (L2R):* This is sent from the TRG to the ROB. On reception of this message the ROB removes and event fragment from its internal buffers.
- *Level 2 Accept (L2A):* This is sent from the TRG to the EBIF. On reception of this message the EBIF collects all the event fragments associated to a single event from the ROBs.
- *Region-Of-Interest (RoI) Request:* This is sent from the TRG to the ROBs. It is a request from the Level 2 trigger system for data. On reception of the message the ROBs output data to the Level 2 Trigger system.
- *Discard:* This is a message sent by the EBIF to the ROBs. It is essentially the L2A message relayed to the ROBs to inform those that data collection for a specific event has been performed and that the associated event fragments may be removed from the ROB buffers.

3. INITIAL IMPLEMENTATION

The initial implementation of the ROC does not provide all the functionality foreseen by the high-level design. External data producers and consumers (e.g. the ROD, the trigger system) are not yet available. Consequently the performance measurements focus on the data flow within the ROC while external data input and output are emulated in different ways.

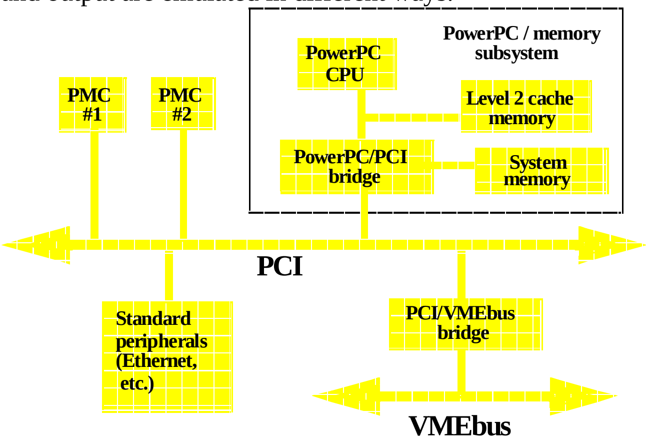


Figure 3: Architecture of the CES RIO2 VMEbus CPU module

The hardware used consists of a multiprocessor system based on the CES RIO2 8061/8062 and the Motorola MVME2600/2300 CPU[1, 2, 3]. The architecture of these modules, see Figure 3, is common to most VMEbus boards based on PowerPC and PCI. An extensive evaluation of the RIO2 and other second generation VMEbus CPU boards[4] have shown that their CPU and I/O (PCI, VMEbus) performances are similar (for the same type of CPU).

The DAQ software runs under LynxOS[5] but the use of operating system specific features was reduced to a minimum in order to obtain a good level of code

[1]http://www.ces.ch/Products/Processors/RIO28060/RIO28060.html

[2]http://www.ces.ch/Products/Processors/RIO28062/RIO28062.html

[3]http://www.mcg.mot.com/WebOS/omf/GSS/MCG/products/boards/vmeppc.html

[4] http://www.cern.ch/ESS/OS/reports/PPC-EVAP.PS

[5]http://www.lynx.com

portability and performance. The OS features still used are:

- Direct access to H/W components from user libraries via shared memory segments
- Reservation of physically contiguous memory
- Time functions
- Exit handling

The TRG, EBIF and ROB applications are single process. For reasons of performance, requests for I/O are served via polling and not by interrupts and I/O drivers. In this implementation all interprocessor communication is via VMEbus and implemented in software as a message passing library based on shared VMEbus memory. The flow of data control messages and event fragments on VMEbus and the associated VMEbus parameters are listed in Table 1. Write posting and read pre-fetching on VMEbus are enabled wherever possible to achieve the best performance. In addition, a single VMEbus request level and fair arbitration were used.

Msg/transfer type	Size (bytes)	Relative frequency	Direction	Transfer type
L2R	24	N^a*100	TRG → ROBs	DMA MBLT D64
RoI	56	N^a*10	TRG → ROBs	Single cycle D32
L2A	24	1	TRG → EBIF	Single cycle D32
Discard	24	N^a*1	EBIF → ROBs	Single cycle D32
DC	N^a*1024	1	ROBs ← EBIF	DMA MBLT D64
Monitoring	N^a*1024	<<1	ROBs → LDAQ	DMA MBLT D64
LDAQ communi-cation	~100	N^b*<<1	LDAQ ↔ IOMs	Single cycle D32

a) Number of ROBs, b) Number of IOMs

Table 1: Data flow on VMEbus

As mentioned above, the initial implementation has reduced functionality in several areas compared to the logical model described in the previous section:

- In the TRG application, data control messages are generated internally. The loading of the PCI bus by an external trigger system is emulated using a simple PMC. The latter transfers data over PCI bus to system memory in DMA mode with a block size of about 1 KByte and at a rate corresponding to that at which control messages are generated internally.
- The ROBs generate event fragments internally and do not transfer any ROB fragments to the Level 2 Trigger system. The RoI requests received do not trigger any data transfers.
- The EBIF performs data collection over VMEbus but does not transfer any event data to the Event Builder.

Since there are no external data sources, the synchronisation between the TRG and the ROBs has to be emulated: whenever a ROB has produced an event it transmits its current event number over VMEbus to the TRG. The TRG will then only send data control messages corresponding to events where all fragments have been generated by the ROBs. The data control messages generated internally by the TRG are produced with the ratio 1 L2A : 100 L2R : 10 RoI.

4. PERFORMANCE MEASUREMENTS

A measure of the global performance of the ROC is indicated by the frequency at which the individual ROBs input event fragments from the detector links. This quantity was measured in a ROC configuration without the LDAQ module, as a function of the number of ROBs in the crate (see Figure 4). The measurements were performed using operating system timing functions and cross-checked using VMEbus analysers from VMEtro[6]. These allow the measurement of the VMEbus transfer rate and bus utilisation.

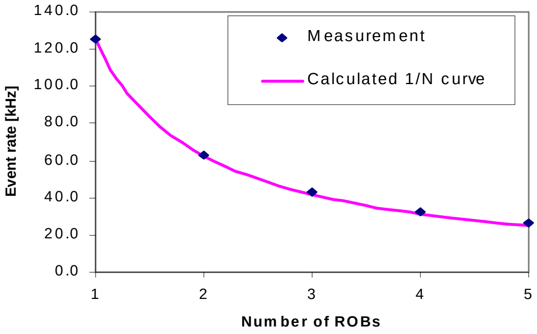


Figure 4: Performance with TRG, EBIF and ROB applications working without the LDAQ control

In the following sections the type of ROB (RIO2 or MVME2604) will not be mentioned explicitly since it has been verified that the results of the performance measurements are not significantly different for the CES and Motorola CPU boards.

For one ROB a performance of 126 kEvents/s has been measured. This value drops to 26 kEvents/s for five ROBs. In the ROC the VMEbus throughput was measured to be 6 Mbyte/s while the VMEbus utilisation was 70% independent of the number of ROBs. A VMEbus utilisation[7] of 70% is not far from the maximum obtainable in a single cycle dominated multi-processor environment[8].

This clearly indicates that, even in a single ROB configuration, the performance is limited by the bandwidth available in a single cycle dominated VMEbus system. The overlaid 1/N curve in Figure 4 confirms this in showing that the product of the number of ROBs and

[6] http://www.vmetro.com

[7] The number of bus samples with BBSY# active divided by the total number of samples

[8] The remaining 30% are consumed largely by arbitration overheads

the number of events processed remains constant. This is because all data control massages are sent sequentially to the receivers. As the messages contain identical information it would be more efficient if they could be broadcast.

Performance measurements have been done for a ROC configuration controlled by an LDAQ module and having only one ROB. In this case the focus was on the effect of event monitoring on overall event rate. One LDAQ sampling transaction with an IOM means in this case an LDAQ monitoring request followed by a DMA transfer of 1KByte event. As Figure 5 shows, an increasing rate of the event sampling by the LDAQ, on different IOM types (ROB, EBIF or TRG) has the effect of degreasing the event rate in the ROC by few percent. The comparison between the three curves shows that the impact of event monitoring is stronger for the ROB than for the other two IOMs. Nevertheless for a event monitoring in the range of 100Hz, the event rate in the ROC has decreased by only ~2 percent in case of ROB monitoring and by 0.4 percent in case of TRG or EBIF monitoring.

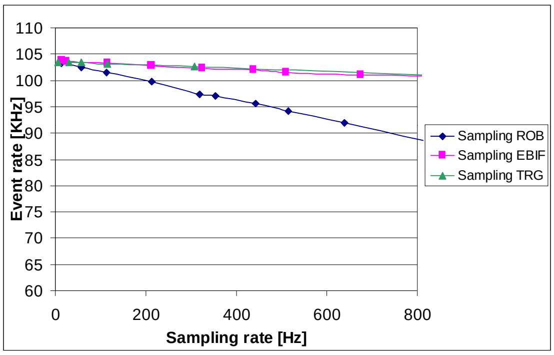


Figure 5: LDAQ impact on the event rate in a ROC configuration: LDAQ, TRG, EBIF, 1* ROB

5. MODELLING

The performance of VMEbus transfers between two RIO2s as well as the characteristics of the on-board PCI have been studied[4]. Based on these measurements and the knowledge of the data flow on VMEbus in the ROC, see Table 1, a model of the ROC (without LDAQ) has been developed to study rate-limiting factors and variations in technologies and architectures. All IOMs (TRG, EBIF, ROBs) of a ROC have been simulated [5] within the discrete event domain of the Ptolemy framework [9]. The model is shown in Figure 6 . The ROD, Level 1, Level 2 and EB are simply modelled as external data producers / consumers. As the measurements done with the ROC have indicated that the overall system performance is limited by the available I/O bandwidth on VMEbus and PCI, software latencies have not been taken into account. Additionally, the transfers on PCI and VMEbus have not been modelled at the protocol level. All external links run at the nominal speed of PCI except for the links to the detector front-end which have infinite bandwidth as the maximum event rate otherwise would be limited by the PCI of the ROBs to less than 70 kHz. The model was parameterised with performance figures measured on the VMEbus processors used and included arbitration mechanisms for both VMEbus and PCI.

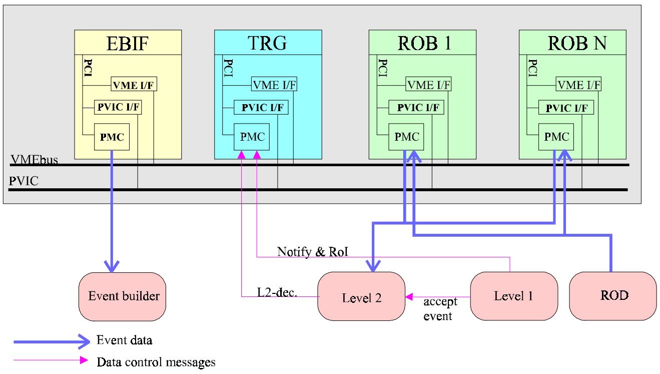


Figure 6: ROC model

The maximum event rate is shown in Figure 7 for five cases:

- Measurements : these are the event rates measured for one to five ROBs and have already been discussed in the previous chapter.
- VME model: this curve was calculated to check the model. It simulates the full intra crate traffic on VMEbus including the synchronisation messages that are sent from each of the ROBs to the TRG (one single cycle per ROB and event). The calculated figures match the measured performance to within a few percent, and therefore confirm that the system is I/O bound and computing times can be neglected.
- VME model w/o synchronisation: this is a simulation of the system performance without the synchronisation messages. These are only required in the prototype to replace the missing external synchronisation (Level 1) but consume an important fraction of the available bandwidth on VMEbus. The curve still shows the 1/N shape of the measurement.
- VME model w/o synchronisation + PVIC: use of two intra-crate busses: VMEbus and PVIC[10]; both connected to the PCI of the IOMs. The model has been parameterised with measured performance figures for

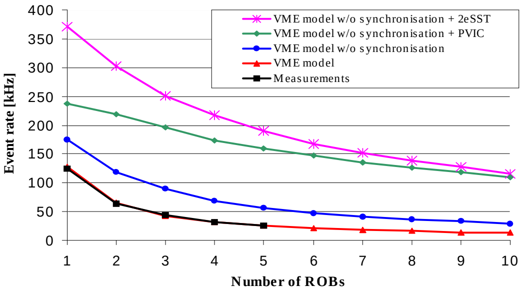
[9] The Almagest – Volume O: Ptolemy 0.7 User’s Manual. http://ptolemy.eecs.berkeley.edu/papers/almagest/user.html

[10] http://www.ces.ch/Products/Connexions/PVICFamily/PVIC.html

PVIC. The intra-crate traffic has been organised such that VMEbus is efficiently used for the data collection (ROBs to EBIF) that mainly consists of D64 MBLT transfers. The PVIC carries all the data control messages. The reject messages from Level 2 are now broadcast to the ROBs as well as the RoI messages and the discard messages. The improvement in performance is mainly due to the broadcast. The additional bandwidth of PVIC and the more efficient use of VMEbus only play a minor role as the total bandwidth is limited by the common PCI bus.

- VME model w/o synchronisation + 2eSST: the 2 edge Source Synchronous Transfer (2eSST) protocol[11] is a proposed addition to the VMEbus standard. Its main features are improved bandwidth (>300 MByte/s) and a broadcast protocol. The system performance has been modelled with all traffic routed across VMEbus. As VMEbus, in the model, still is routed via PCI, the PCI bandwidth was adjusted to the figures for a 64bit / 66 MHz PCI (528 MByte/s) in order to be able to exploit the full potential of the 2eSST protocol. The comparison to the curve "VME model w/o synchronisation" demonstrates the effects of the 2eSST protocol. The higher bus bandwidth globally speeds up the system by a factor of three and the additional broadcast capability flattens the curve, which leads to an even better performance for ROCs with many ROBs.

Figure 7: Maximum event rate (measurements and simulations)



It is to be noted that the model has been parameterised with (conservative) speculative figures for the timing of a 2eSST capable VMEbus processor.

6. SUMMARY AND CONCLUSIONS

The Read-Out Crate in ATLAS DAQ/EF prototype -1 has been described. A configuration consisting of a LDAQ, TRG, EBIF and up to five ROBs has been implemented in VMEbus using CES RIO2 8061, RIO2 8062 and Motorola MVME2604 PowerPC/PCI based processor modules. Data control and synchronisation messages between IOMs are exchanged via VMEbus, which is also used for the Data Collection. External input to the TRG is emulated by a simple PMC interface, while the ROBs and the EBIF have no external I/O. The performance was measured in terms of the event rate seen by the ROBs.

The performance results can be summarised as follows:

- With one and five ROBs the performance was measured to be 126 kHz and 26 kHz, respectively. The performance decreases inversely to the number of ROBs.

- The performance was limited by the rate at which data control messages could be transferred over VMEbus. A throughput of only 6 Mbyte/s was measured and is due to the use of VMEbus single cycles in the message passing and the method of synchronisation between the TRG and the ROBs.

A simple discrete event model (PTOLEMY) of the data flow on both the PCI of the IOMs and VMEbus has been developed. It has allowed to quantitatively verify that the global performance of the ROC is dominated by VMEbus I/O. Based on measured and estimated values for two broadcast capable buses, PVIC and VMEbus 2eSST, the system performance has been computed. The results of the model are:

- There is a good agreement between the measurements and the model for the current implementation.

- The model has shown that the additional synchronisation messages, required due to the absence of an external trigger, do influence the overall system performance at the level of 20-30%.

- Adding a secondary, broadcast capable bus (e.g. PVIC) to the system significantly improves the performance by a factor of ~3 to 240 kHz for one ROB and 110 kHz for 10 ROBs respectively.

- Routing all traffic via an improved VMEbus based on the proposed, broadcast capable 2eSST protocol leads to a speed-up slightly higher than that calculated for the dual bus scenario with the additional advantage of a less complex system.

The results presented above, obtained in a complex multiprocessor DAQ application, confirm some observations made previously[4]. Modern PowerPC/PCI VMEbus CPU boards have shown impressive progress in the area of processing but much less so in the area of I/O, notably VMEbus. This may change with the advent of new VMEbus standards (VME64x and 2eSST).

The results also show that a system based on COTS components is a viable solution. Even though the maximum performance obtainable with today's hardware is still significantly below the final ATLAS requirements we are confident that this design in combination with forthcoming technologies can provide the required performance on the time scale of ATLAS.

[11] VITA 1.5, 2eSST, draft 1.5, http://www.vita.com/vso/draftstd/2eSSTd1.5.pdf

7. REFERENCES

[1] The ATLAS Collaboration, Technical Proposal for a General Purpose pp Experiment at the Large Hadron Collider at CERN. CERN/LHCC/94-43.

[2] G. Ambrosini et. al., The ATLAS DAQ and Event Filter Prototype "-1" Project, presented at Computing in High Energy Physics 1997, Berlin, Germany. http://atddoc.cern.ch/Atlas/Conferences/CHEP/ID388/ID388.ps

[3]G.Ambrosini et al., Event Building in the ATLAS DAQ/EF Prototype "-1", presented at Computing in High Energy Physics 1998, Chicago, USA. http://www.hep.net/chep98/234.html

[4]C.P.Bee at al., The ATLAS Event Filter, presented at computing in High Energy Physics 1998, Chicago, USA http://www.hep.net/chep98/225.html

[5]B. Rensch, ROC simulation with Ptolemy, DAQ/EF Prototype –1 Technical Note 90.

[6] http://atddoc.cern.ch/Atlas/Notes/041/Note041-1.html