

A Prototype ROI Builder for the Second Level Trigger of ATLAS Implemented in FPGA's¹

R. E. Blair, J. W. Dawson, W. N. Haberichter, and J. L. Schlereth, Argonne National
Laboratory

M. A. Abolins and Y. Ermoline, Michigan State University

Abstract

In an effort to reduce data transfer and rate requirements, the Higher Level Trigger of the ATLAS Detector uses Region of Interest (ROI) information forwarded from Level 1 Partitions on a Level 1 Accept. The ROI Builder receives these ROI fragments, which may be considerably skewed in time and may be interspersed with fragments from other events, organizes and formats from these fragments a record for each event accepted by Level 1, selects a processor to manage the event, and transfers via S-link the assembled ROI record to the target processor. The ROI Builder must fulfill these requirements at the Level 1 Trigger rate of 100 kHz while accommodating S-link flow control. A design for the ROI Builder was developed emphasizing parallelism, implemented in FPGA's, and has been run in Testbeds at Saclay and CERN.

1. Overview

The Level 2 Trigger of ATLAS is envisioned as a multiplicity of processors connected by a high-speed switching network. Suitable candidates for this network might be giga-ethernet or ATM. Also connected to this network are the Readout Buffers (ROB's) which receive the global event data from the Front End Electronics, and the Supervisor, which manages the operation of the Level 2 Trigger. To minimize data transfers on the network, the Level 2 Trigger uses ROI information from Level 1. This information is transferred to Level 2 on a Level 1 Accept in the form of ROI Fractions. Only data relevant to ROI's is used in the Level 2 trigger algorithms.

ROI information is transferred from a number (currently 7) of Level 1 processors to the Level 2 Trigger Supervisor as ROI Fractions. The ROI Fractions are received within the Supervisor by the ROI Builder. The ROI Builder must build an ROI Record by assembling the ROI Fractions corresponding to each event, allocate the event to an ROI Processor within the Supervisor, and transfer the ROI Record to the target ROI Processor. The selected ROI Processor manages the event through Level 2. It allocates the event to a processor on the Switching Network, forwards the ROI Record to this target processor, and waits for a Level 2 Accept or Reject.

¹ Based on presentation by J. W. Dawson at the 5th Workshop on Electronics for the LHC Experiments (LEB 99), Snowmass, CO, USA, 20 – 24, Sep. 1999.

2. Objectives of this work

This prototype development is intended to demonstrate the feasibility of a candidate architecture capable of building ROI Records at the maximum Level 1 Trigger rate of 100 kHz without introducing dead time. Another objective of this work is to provide hardware support to the Pilot Project within the ATLAS Level 2 Trigger Group. The Pilot Project is a hardware/software effort to emulate subsets or slices of a particular ATLAS Level 2 Trigger architecture and represents a continuation of the Demonstrator Program of last year. Additionally, we hope later this year to be able to integrate the ROI Builder with Level 1 Trigger elements at RAL and DAQ-1 prototype hardware at CERN.

3. Design Philosophy

This prototype must receive a variable number of ROI Fractions per event via S-Link from the Level 1 Processors, must provide assembled ROI Records via S-Link for as many as 8 ROI Processors, and must operate without introducing dead time at rates as high as 100 kHz. In order to meet these requirements, we have implemented the ROI Builder entirely in hardware, and have emphasized parallelism in the design. Our selection of hardware has been the Altera 10K family of FPGA's. As implemented to serve 8 ROI processors, the ROI Builder uses 76 10K40's and 8 10K50's. This extremely dense logic allows us to use an architecture where essentially we have 8 ROI Builders operating in parallel, connected only by the event selection algorithm (Round Robin in this implementation). The event selection algorithm, implemented in hardware, uses the Level 1 Event ID embedded in every ROI Fraction to identify the ROI Fractions belonging to a given event.

The ROI Builder is tolerant of ROI Fractions which are skewed in time by as much as 100 μ sec, which may arrive asynchronously, and which may not preserve the time-ordering of the fragments. The ROI Record is built in 2 μ sec after receipt of the last fragment of an event, and is immediately transferred via S-Link to the target ROI Processor. As implemented, each ROI Builder card can build ROI Records from as many as 12 input data streams, and can service 2 ROI Processors. The ROI Builder is completely data driven and

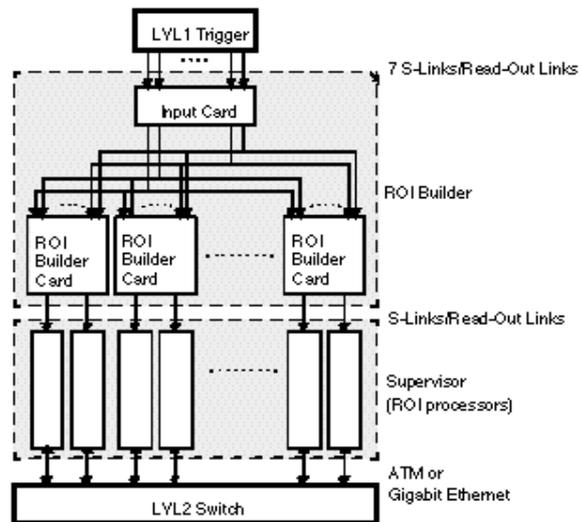


Figure 1: Block Diagram of ROI Builder

scalable. Accordingly, a Prototype ROI Builder using 4 ROI Processors uses 2 ROI Builder Cards, and a Prototype ROI Builder using 8 ROI Processors requires 4 ROI Builder Cards.

Figure 1 shows the structure of the Supervisor in block form.

To improve rate capability and avoid bottlenecks, the logic design exploits parallelism as much as possible. Each input ROI Fraction is received simultaneously by the input buffer of each channel of ROI Builder logic, and the Level 1 Event ID is checked. The channel for which this number is relevant passes the Fraction to a second buffer where the record is built. All channels for which the Level 1 Event ID is not relevant immediately clear their input buffer and are ready for another Fraction. For this prototype, events are allocated on a round robin basis, however for later versions of this hardware much more sophisticated algorithms are possible.

Figure 2 shows one of the prototype ROI Builder Cards. There are 12 inputs for data streams from Level 1 carried via copper in S-Link format, six input FPGA's which

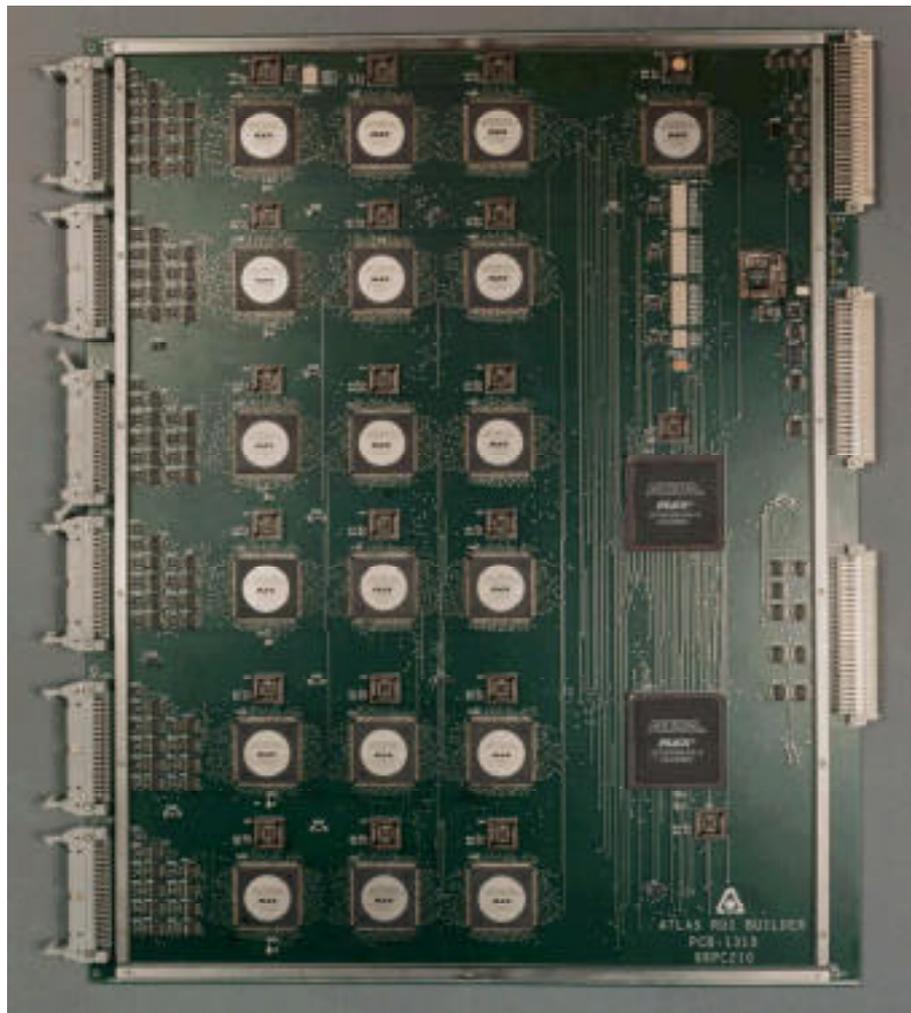


Figure 2: Prototype ROI Builder Card.

are configured as 12 input buffers, 12 FPGA's which provide the two secondary buffers in which the records are built, and two FPGA's on pin grid arrays which manage transfer of the assembled records to the two target ROI Processors served by this card.

4. ROI Builder Tests

To facilitate diagnostic testing of the ROI Builder and for use as a Level 1 emulator in the test beds, an input card was designed and built. This input card emulates six Level 1 processors, and furnishes to the ROI Builder the ROI Fractions, which would be transferred on Level 1 Accepts. This input card can be loaded from VME with ROI

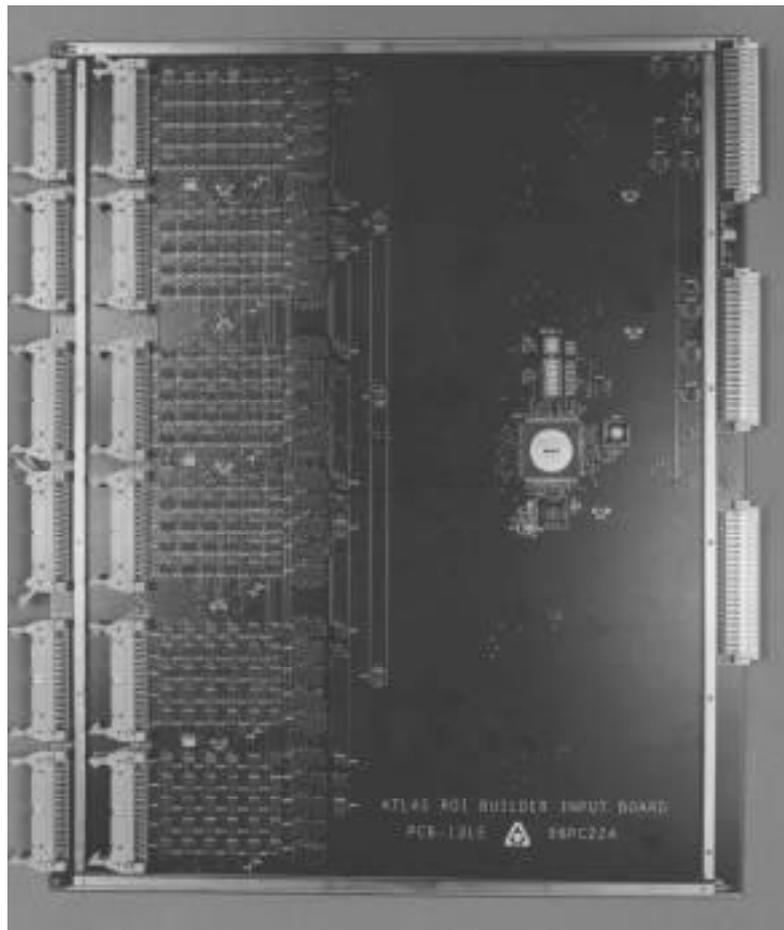


Figure 3: Input Card.

Fractions for 1024 events, and these events can then be initiated at one of 12 software selectable rates, or under VME control. This input card implements flow control precisely as S-Links from Level 1 Processors would. The input card is shown in Figure 3.

Diagnostic codes were written in C++ which were capable of exercising all the functions of the ROI Builder. This self-contained program running under LynxOS on the CES RIO2 provided a menu-driven interface and had functions to:

- Set data clock (4 settings, 33-264 ns period).
- Set event clock (12 settings, 4-100kHz).
- Load and check event memory (1024 events).
- Set running mode, continuous at preselected rate or VME selected.
- Read out one S-Link port in either DMA or single word mode.
- Compare received with expected data and log errors.
- Measure actual event rate.

After testing the hardware, the system was set up and a series of tests was run. The test system components comprised:

- One input card emulating 6 Level 1 partitions
- Two ROI Builder cards
- Two S-Link output cards (4 S-Link output channels)
- Four i686 PC's running under Linux or 4 RIO2's running under LynxOS
-

The purpose of this testing was to make sure the system ran without errors, and to investigate the Level 1 trigger rates that could be supported without having the Supervisor create deadtime. In these measurements, 1, 2, and 4 nodes were used. The input was run freely at the 12 pre-selected rates with S-Link flow control implemented. The event rate was measured both for S-Link transfer only and for S-Link transfer and data unpacking where the system output was checked for errors. Figure 4 shows the experimental setup, and Figure 5 shows typical data from the testing.



Figure 4: The Experimental Setup.

5. Integration at Saclay

In March 1999 the Supervisor/ROI Builder was integrated into a 32 node ATM network at Saclay and a series of tests were run. The goals of this work were to operate

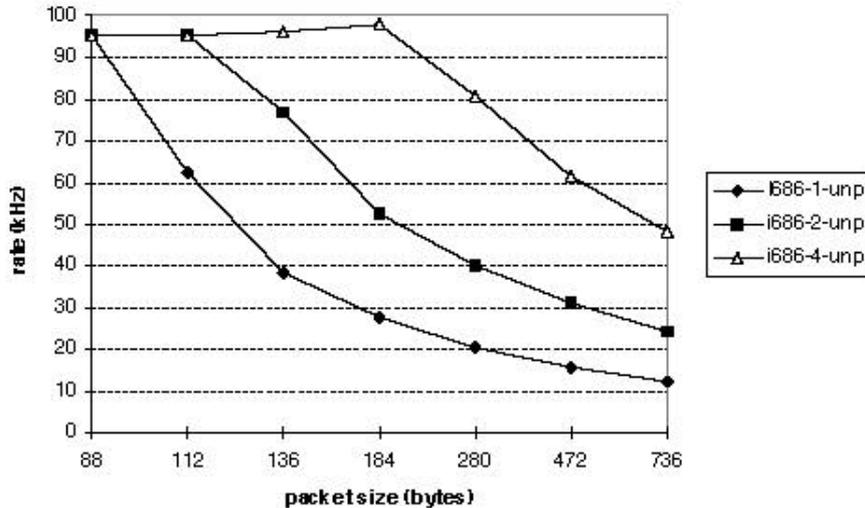


Figure 5: Data Unpacking Rate for 1, 2 or 4 i686 Nodes.

the ROI Builder in a 32 node network using 1, 2, or 4 Supervisor nodes, to find the limits of Supervisor performance using existing processors (200, 300 MHz RIO2's), and to investigate the effect that S-Link flow control may have on the performance. Figure 6 is a conceptual view of the setup. Extensive testing was conducted, and a brief summary of results is presented in Table 1.

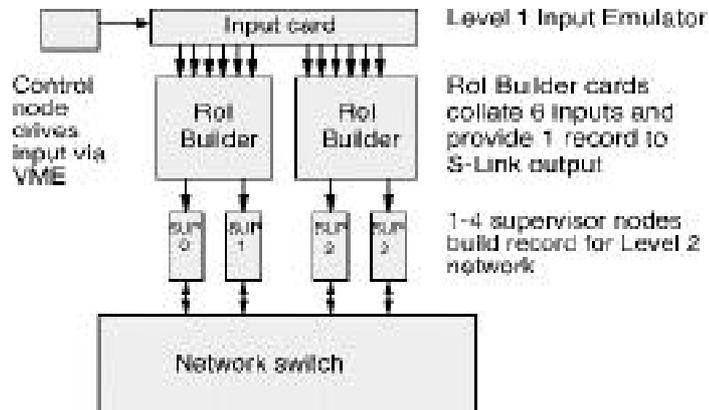


Figure 6: ROI Builder - ATM Integration.

Table 1: Summary of Results

Configuration	Maximum Rate (kHz)	$\mu\text{sec/event}$
ROI Builder Readout	67	15
1 Supervisor Emulator	39	26
1 Supervisor – ROI Builder	29	34
2 Supervisor – ROI Builder	$27 + 28 = 55$	18
4 Supervisor – ROI Builder	$23 + 23 + 24 + 24 = 94$	10.6

6. Conclusions

A prototype ROI Builder for ATLAS Level 2 Trigger has been built using FPGA's in a highly parallel architecture. It has been integrated in a 32-node ATM network at Saclay, and has been found to satisfy the requirements for the Level 2 Trigger of ATLAS. This prototype hardware has now been moved to an Ethernet test bed at CERN for further evaluation and testing.

The most difficult problems encountered in this work related to maintaining data integrity in the presence of flow control constantly turning on and off. Flow control tends to be raised in the S-Link connections from the ROI Builder to the ROI Processors, because the ROI records are typically longer than 128 words. If the ROI Processor does not promptly service the PCI port, then flow control will ripple back through the ROI Builder into the input S-Links. Although there is a considerable amount of buffering, eventually the input S-Links will be blocked.