Pre-Production Validation of the ATLAS Level-1 Calorimeter Trigger System

R. Achenbach[1], C. Ay[2], B.M. Barnett[3], B. Bauss[2], A. Belkin[2], C. Bohm[4], I.P. Brawn[3],
A.O. Davis[3], J. Edwards[3], E. Eisenhandler[5], F. Föhlisch[1], C.N.P. Gee[3], C. Geweniger[1],
A.R. Gillman[3], P. Hanke[1], S. Hellman[4], A. Hidvégi[4], S.J. Hillier[6], E.-E. Kluge[1], M. Landon[5],
K. Mahboubi[1], G. Mahout[6], K. Meier[1], A. Mirea[3], T.H. Moye[6], V.J.O. Perera[3], W. Qian[3],
S. Rieke[2], F. Rühr[1], D.P.C. Sankey[3], U. Schäfer[2], K. Schmitt[1], H.-C. Schultz-Coulon[1],
S. Silverstein[4], R.J. Staley[6], S. Tapprogge[2], J.P. Thomas[6], T. Trefzger[2], D.Typaldos[6],
P.M. Watkins[6], A. Watson[6], G.A. Weber[2], P. Weber[1]

[1]Kirchhoff-Institut für Physik, University of Heidelberg, Heidelberg, Germany
[2]Institut für Physik, University of Mainz, Mainz, Germany
[3]CCLRC Rutherford Appleton Laboratory, Oxon, UK
[4]Fysikum, University of Stockholm, Stockholm, Sweden
[5]Physics Department, Queen Mary, University of London, London, UK
[6]School of Physics and Astronomy, University of Birmingham, Birmingham, UK

Abstract- The Level-1 Calorimeter Trigger is a major part of the first stage of event selection for the ATLAS experiment at the LHC. It is a digital, pipelined system with several stages of processing, largely based on FPGAs, which perform programmable algorithms in parallel with a fixed latency to process about 300 Gbyte/s of input data. The real-time output consists of counts of different types of trigger objects and energy sums. Prototypes of all module types have been undergoing intensive testing before final production during 2005. Verification of their correct operation has been performed stand-alone and in the ATLAS test-beam at CERN. Results from these investigations will be presented, along with a description of the methodology used to perform the tests.

I. INTRODUCTION

The ATLAS Level-1 trigger has to provide a decision within a fixed time of 2 µs in order to reduce the LHC bunch-crossing rate of 40 MHz down to a rate of less than 75 kHz of events to be retained for the second level of event selection. The Level-1 decision is based only on reduced granularity calorimeter and muon detector data. The ATLAS Calorimeter Trigger is the part that processes the calorimeter information, which consists of over 7000 analogue signals. The algorithms used have to be simple enough to be performed over a large number of input signals in this limited time, but sophisticated and flexible enough to distinguish potentially interesting particle signatures from a large and, to some extent, unpredictable background.

These requirements have necessitated a design which incorporates several layers of processing being performed in parallel, with all algorithms implemented in FPGAs to allow flexibility. The nature of the algorithms, which make extensive use of overlapping, sliding windows, mean that the ability to transfer large amounts of digital data between modules is a crucial aspect of the system. Testing the correct performance of the algorithms and the stability of the many high-speed links needed forms the basis of the validation of the system and module design before final production takes place in 2005.

The system consists of several designs of module, and full specification prototypes exist for each of these. Many types of tests have been performed on the modules individually, and when working together in a full slice through the processing chain. They were also deployed successfully at the ATLAS test-beam in 2004 when connected to calorimeter trigger signals. The methodology of these tests will be described, along with some of the results.

II. THE ATLAS LEVEL-1 CALORIMETER TRIGGER ARCHITECTURE

The real-time output of the trigger system consists of counts of electron/photon-like, tau-like, or jet-like clusters above programmable transverse energy thresholds, as well as results of threshold comparisons on missing and total transverse energy to be sent to the Central Trigger Processor (CTP). However, all of the modules also have read-out capability, in order to verify the correct performance during normal operation. This read-out only occurs on events which pass the CTP Level-1 decision. On these events, additional location information on trigger objects (known as Region-of-Interest data) is also sent to the Level-2 trigger system.

The basic architecture of the whole of the Level-1 system was documented in an ATLAS TDR in 1998 [1]. Some evolution of the calorimeter trigger has taken place since then and a detailed description of the current design has been presented in [2]. A brief outline is given here as background

to the testing environment. A simplified schematic of the modules and dataflow is shown in fig. 1.

The real-time path consists of three subsystems: the Preprocessor (PPr), Cluster Processor (CP) and Jet/Energy-sum Processor (JEP). The Preprocessor system consists of 124 Preprocessor Modules (PPM), which provide the input data used by both the CP and JEP systems. They take analogue pulses, mostly corresponding to 0.1x0.1 sums in eta/phi space, from the ATLAS calorimeters, digitize and synchronize them, and identify the bunch-crossing from which each pulse originated. Finally, lookup tables perform the E_T calibration for these trigger towers which form the basis of the trigger. Data are sent downstream to the CP and JEP systems using LVDS 400 Mbit/s serial link chipsets in order to reduce the I/O requirements on cables and pins.

The Cluster Processor system consists of 56 Cluster Processor Modules (CPM) which identify and count electron/photon and tau candidates. The final sums are performed in 8 Common Merger Modules (CMM), and sent to the CTP. The Jet/Energy-sum processor (JEP) consists of 32 Jet/Energy Modules (JEM) which count jet candidates and make missing and total transverse energy sums, with the final results again being summed in 4 CMMs. Both systems require the exchange of a large volume of data between neighbouring modules, for which a common custom backplane has been designed. This backplane contains over 15,000 pins, through which digital signals with speeds of up to 400 Mbit/s differential and 160 Mbit/s single-ended are propagated.

The read-out and Region-of-Interest data is handled by 20 Readout Driver modules (ROD). These receive signals from all of the other modules via optical links running at a maximum of 800 Mbit/s using the Agilent G-link protocol. The data is reformatted into standard ATLAS event fragments, and transmitted on optical links using the ATLAS S-Link protocol.

From this description, it can be seen that the difficult task of extracting trigger objects is achieved by making use of a highly parallel, multi-stage processing design. In order to red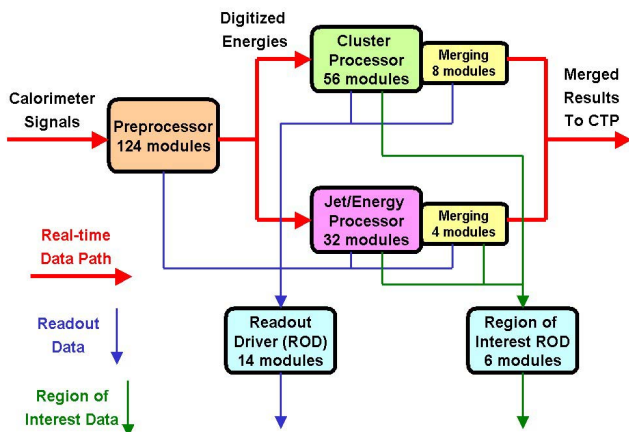uce the number of different kinds of modules needed, multi-purpose modules have been designed, utilizing the flexibility of the FPGAs on which the system is based [3].

III. GENERIC MODULE DESIGN AND CHALLENGES

There are five different types of module required for the main task of the trigger, and each of these conforms to a common design. In general they consist of several stages of processing, and each stage is performed by a bank of one or more FPGAs working in parallel. Thus the detailed module design mirrors the system design in being made up of several stages of parallel processors. This generic module design is illustrated in fig. 2. Note that cable inputs and outputs are routed via the backplane wherever possible to facilitate module insertion and extraction. The only module that is significantly different from this model is the Preprocessor, which is constrained by the analogue nature of its input, and makes use of an ASIC rather than an FPGA as its key processing element.

There are several critical elements in the module designs, and testing these forms the most important part of the module validation. Firstly, the FPGA algorithms used to identify trigger objects have to be quite sophisticated, and correct operation must be verified throughout the many FPGAs in the system. Secondly, the high data rates into and out of each module and individual component mean that signal speeds have to be fast, both internally across the module and between modules on cables and via the backplane. Measuring the integrity of these signals is a key goal of the test programme. Finally, once each individual module type is thought to be working, then it is necessary to integrate all the modules into a slice through the full trigger system, and verify that the whole system performs as expected. This can be performed in the
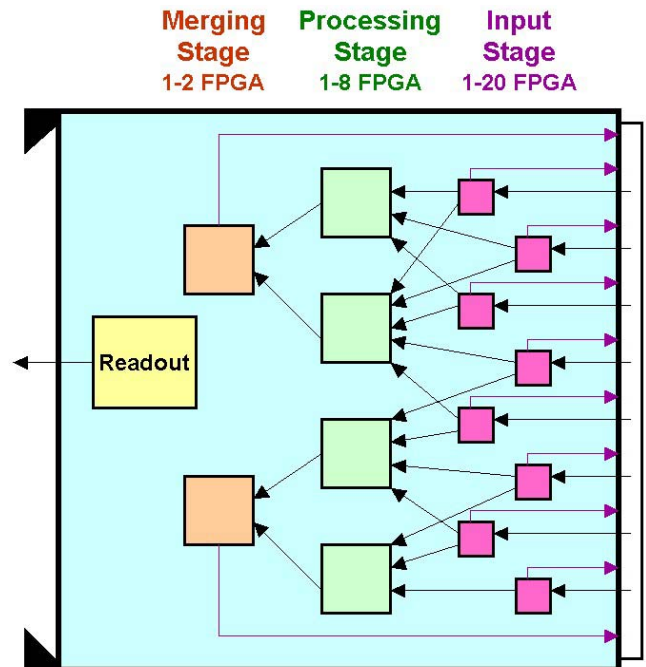

Fig. 2. Generic module design showing some data paths.
Module front panel is at left, backplane at right

Fig. 1. Module types, numbers and connectivity in the complete Level-1 Calorimeter Trigger System

laboratory, but more importantly was also done at the ATLAS combined test-beam in 2004.

Tools are built into the architecture to allow for testing at many points. Typically, an FPGA design will have a method of injecting test data into the system from a playback memory, and also a spy memory to look at the results of processing. The readout data, which provides information on event processing at several points along the chain, is also essential for monitoring performance during normal running conditions.

IV. LABORATORY TESTING METHODOLOGY

Many different levels of testing have to be performed on each module, starting from the initial JTAG and power-up tests, through to final integration. Some of the higher-level tools that have been developed for these purposes are described below. These techniques can be applied both to individual modules and to tests of an integrated system.

A. Simulation Software

The task of identifying a problem with data integrity or algorithm performance is made difficult by the nature of the processing. For example, in several instances throughout the system, data is encoded or compressed in a non-trivial way. Also the final trigger bits, though small in number, are a product of algorithms which are difficult to calculate. These issues, and the size of the system, necessitated a more automatic and systematic approach to checking the correctness of data passing through the system.

This was achieved by building a simulation framework, in which all the modules in the system could be modeled at the level of data that can be fed into, or read out of the module. The simulation framework is a C++ library [4] which was inspired by VHDL. It contains similar concepts of input and output data ports, and entities which perform processes on their inputs. It is also designed in a hierarchical way, such that any useful group of processes (e.g. those that make up an FPGA, or a whole module) can be encapsulated and duplicated in an simple way.

The basic framework has been used to build models of the trigger modules, and these models have been further integrated into the standard ATLAS online software environment [5]. Thus the simulation can be controlled from the ATLAS run control, and configured from an online database. For any run, this database specifies which modules are present, how they are configured, cabled and what test-vectors should be used to fill them.

The advantage of this full integration into the ATLAS online software environment is that when the hardware is started via the standard mechanism, all of the hardware configuration information on modules and cabling is also available to the simulation in order for it to predict what the resulting data should be. Typically the simulation produces the expected readout information that should be produced if the trigger is working correctly, for direct comparison with the data read out through the standard data acquisition path.

However it is also often used to predict spy memory contents for more detailed, low-level checks.

B. Artificial Trigger Generation

In order for the simulation to predict readout contents, it has to know when a trigger was generated. This is possible if some of the trigger output bits are available to form a Level-1 accept, but often tests are done without the merger modules which produce the final results. Also, using that method to trigger events would mean that events that did not form a positive trigger decision could not be tested. Instead an artificial trigger generation technique is used, where the exact timing of each trigger generated can be controlled and therefore the simulation also knows when to generate readout data.

This trigger generation has to be synchronized both to the clock used to drive the trigger modules, and to the playback data cycling in the module FPGAs. This is done using a custom module which is capable of generating known patterns of triggers lasting up to 5 seconds, outputting bits from a playback memory. It is clocked by the system clock, and started via a broadcast command which synchronizes all the playback memories in the system. The patterns that can be generated range from simple, equally spaced triggers at a user-defined rate, to more demanding patterns where triggers arrive in closely spaced bursts, down to the ATLAS defined minimum of 125 ns apart.

One further useful feature of the trigger patterns is that they can be optimized both for test vector coverage and simulation speed. Simulating 5 seconds (i.e. 200 million events) of system processing would take too long to be of practical use, since the simulation is run in real-time at run start. However, a simple optimization technique is used. Typically, the playback memories in most of the modules have a length of 256 words. If the triggers are spaced by $(256*n+1)$ clock periods, then the results are the same as if the triggers followed each other, since the playback wraps and repeats after 256 events. The most optimized trigger patterns have all the triggers spaced in this way, so that for the 256 events in a typical playback memory, only 256 cycles must be simulated, rather than the full trigger pattern period. This also has the advantage that if no triggers are vetoed by busy logic, the pattern of events sampled follow the playback memory sequence exactly, and only 256 events are needed to verify the full sequence, before moving onto another set of test-vectors.

C. Signal Strobing Timing Windows

Signals arriving at an FPGA on a module can have their timing affected by several things: the output strobe of a source module, cable length, or signal path length on the PCB. While in a fully parallel system this would not be a problem, there are many points in the trigger system where data from several sources must be synchronized and processed together. The process of strobing and re-synchronizing several input signals in, e.g., an FPGA performing a trigger algorithm is one of the

more challenging features of the system, given the speed of some of the signals involved (up to 160 Mbit/s).

In order to establish good timing windows for the input strobes, the techniques described above (using simulation/hardware comparison) can be used in conjunction with scanning the strobe timing, typically over the 25 ns period of the system clock. Histograms are made of errors against timing, and safe values for the strobe can then be derived. A typical plot, for 40 Mbit/s input into the 20 CPM input FPGAs, can be seen in fig. 3, where the black regions indicate the data mismatch timings.

The error-free window size over the whole module for this strobe is about 20ns, where the distribution of track lengths to the 'zigzag' spaced FPGAs has a large effect on the size of the window. The same is true for the error-free windows of the main processor FPGAs of the CPM and JEM modules, where the signal speeds are 160 Mbit/s and 80 Mbit/s respectively, and the signals are arriving both internally from FPGAs in the same module and from modules on either side in the crate. Here, path length differences become even more significant compared to the signal period. The situation for these processor FPGAs is summarized in table 1, where the number of input pins driven is also given.
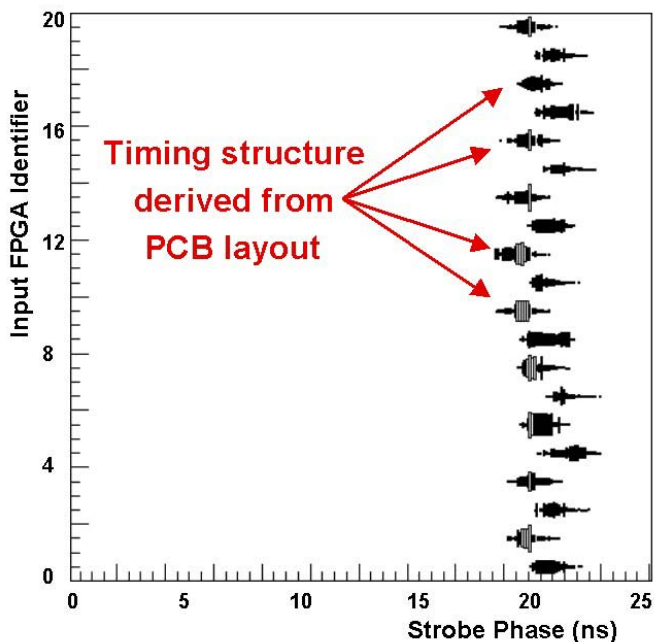

Fig. 3. Timing Window measurement for CPM input FPGAs

TABLE 1
PROCESSOR FPGA INPUT STATISTICS

Module	Number of Pins	Speed Mbit/s	Period ns	Timing Window, ns
CPM	108	160	6.25	2.5
JEM	385	80	12.5	8.0

D. High Statistics Real-Time Link Testing

Using software comparison, it is only possible to test a limited number of events (typically 100 Hz). While this is sufficient for testing algorithms, it is insensitive to low-level bit-error problems in the high-speed links. Other techniques are therefore employed. Firstly, data words throughout the system are protected with a single odd-parity bit. Typically 8-10 bits of data have a parity bit appended, but other, larger, groups of bits are also protected on lower-speed links. This is the simplest possible data corruption detection possible, but it does guarantee that single bit errors will be detected. More sophisticated encoding is prohibited by bandwidth requirements. Parity-error data is flagged and counted, so if a link is unstable it can quickly be detected. Several modules have been run overnight with no parity errors seen, implying a bit-error rate of less than 10^{-14}.

To extend these measurements, dedicated firmware loads have also been used to check incoming data in real time. Known patterns of data are sent along a link, and the receiving FPGA has a firmware variant that knows exactly what data to expect. Once synchronized to the input data, the firmware counts any errors seen. This type of test has been run for 15 minutes with all JEM inputs loaded, and no errors were seen. Even in this short time, this sets an upper limit of 10^{-13} on the bit-error rate, which is far better than is needed for the trigger system.

V. PERFORMANCE AT THE ATLAS TEST-BEAM 2004

While laboratory results were encouraging, the ultimate test of the trigger system was to demonstrate that it could perform in a genuine physics environment. This was possible at the ATLAS combined test-beam at CERN in 2004. Both the Liquid Argon (Electromagnetic) and Tile (Hadronic) Calorimeters were present, and they provided the trigger system with summed towers of data as in the final system. The Level-1 Calorimeter Trigger successfully integrated into the test-beam infrastructure, providing readout for comparison with the detectors, and also briefly providing a trigger via the CTP, which successfully identified high energy events in the detectors.

The trigger hardware was comprised of a slice through the whole system, with one PPM, one JEM, one CPM, two CMMs along with several RODs to format all the necessary data types. The setup, and some of the early results, are described elsewhere [6]. Some more recent results are presented below. These take two main forms, comparison of the trigger readout with other detectors, and internal checks of the trigger data.

A. Detector Correlations

The test-beam provided a unique opportunity to integrate with the calorimeters, and one of the most important results was the correlation seen between the detector energy reconstruction and the energies as seen in the trigger towers read out from the trigger hardware. This was despite the fact that there was little time to establish the exact timings and

correct filter coefficients. Fig. 4 shows scatter plots of the correlations between the two calorimeters and the calorimeter trigger.

The results with the hadronic calorimeter are the most encouraging – the saturation effect seen at about 230 GeV in the trigger energies is a known consequence of the simple filter algorithm used in the PPM for the test-beam period. This will not happen in the final system with correctly matched filter coefficients and proper energy calibration. The correlation with the electromagnetic calorimeter is less precise, with some events entirely lost. There is some evidence that this was due to a misidentification of bunch-crossing for some events due to particles arriving close together in time.

B. Internal Consistency Checks of Trigger Data

The trigger readout data consisted of intermediate results from the CPM, JEM and CMMs. Several cross-checks were possible within this entirely digital data. Both the CPM and JEM recorded the incoming energies, so these could be compared. They were found to be identical in all respects, confirming the stability of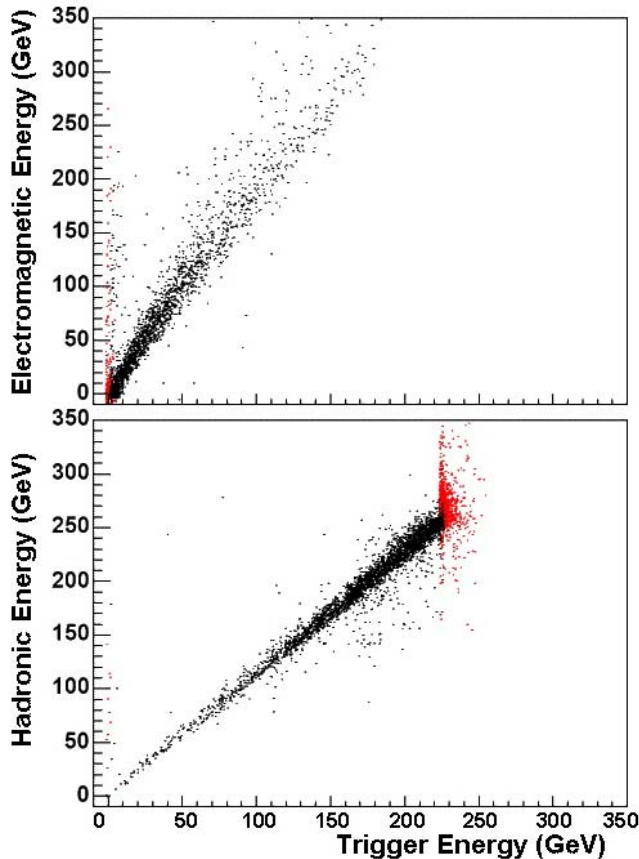 the populated digital links. From the incoming energies, the results of the various physics algorithms – electron clusters, jets and energy sums – could be predicted. These were checked against those recorded in the data both by the processing modules and the CMMs, and again there was no evidence of data corruption. Some minor algorithmic problems were seen, but these were understood as firmware bugs, which could be easily fixed.

For the 0.5 million events recorded while the trigger was active, no significant problems were found in the digital processing. It should be noted, however, that while this was a complete slice through the system, it was very much reduced in scale (32 input signals compared to over 7000 in final ATLAS).

VI. CONCLUSION

A small number of prototype full specification modules for the Level-1 Calorimeter Trigger System have been tested thoroughly, both in the laboratory and in test-beam conditions, and found to work well. The set of tests used are sophisticated and well developed, and should be invaluable in the production testing and installation phase of the project, which will begin during 2005.

REFERENCES

[1] ATLAS Level-1 Trigger Group, *ATLAS First Level Trigger Technical Design Report*, ATLAS TDR-12, CERN/LHCC/98-14, CERN, Geneva,1998
[2] J.Garvey et al., *The ATLAS Level-1 Calorimeter Trigger Architecture*, IEEE Trans. Nucl. Sci., Vol. 51, p356-360, 2004
[3] I. Brawn, *One Size Fits All: Multiple Uses of Common Modules in the ATLAS Level-1 Calorimeter Trigger*, 7th Workshop on Electronics for LHC experiments, CERN/LHCC/2001-034, 22 October 2001,p.253
[4] Simulation Framework information can be found at: http://hepwww.ph.qmul.ac.uk/l1calo/sweb/documents/doclist.html
[5] I. Alexandrov et al.. Online Software for the ATLAS Test Beam Data Acquistion System, IEEE Trans. Nucl. Sci., Vol. 51, p578-584, 2004
[6] J. Garvey et al., *Beam Test of the ATLAS Level-1 Calorimeter Trigger System*, presented at IEEE-NSS, Rome, 2004

Fig. 4. Calorimeter energies compared to energy as seen by the trigger