The ATLAS liquid argon calorimeters ReadOut Drivers. A DSPs and FPGAs based design

J. Prast

LAPP-TECH 2003-02
April 2003

The ATLAS Liquid Argon Calorimeters ReadOut Drivers A DSPs and FPGAs based design

J. Prast

LAPP-IN2P3-CNRS

BP. 110 – F-74941 Annecy-le-Vieux Cedex

Presented at the International Signal Processing Conference ISPC,
Dallas (USA), March 31 - April 3, 2003.

The ATLAS Liquid Argon Calorimeters ReadOut Drivers
A DSPs and FPGAs based design

Julie PRAST
LAPP/CNRS
Chemin de Bellevue
74940 Annecy le Vieux
tel : 00 33 450 091 787
Julie.prast@lapp.in2p3.fr

ABSTRACT

ATLAS is one of the two experiments of the Large Hadron Collider (LHC) at CERN (Geneva-Switzerland), which will study proton-proton collisions at a center of mass energy of 14 TeV. The Liquid Argon calorimeter, which is one of the main detector of ATLAS, is tailored to perform accurate electron, photon and hadron identification, energy, position and time measurements.

The ReadOut Driver (ROD) is the key element of the ATLAS Liquid Argon Calorimeter readout system. It calculates the precise energy deposited in each calorimeter cell and the timing of these signals from discrete time samples. This is done by applying an optimal filtering algorithm, in order to minimize the background noise contributions. It also performs monitoring and formats the results for the next element in the electronic chain.

Each ROD module receives selected data from 1024 calorimeter cells at a maximum rate of 100 kHz. It consists of a 9U VME motherboard, into which are plugged 4 processing daughterboards. While the motherboard main task is reception, distribution and transmission of the signals, the daughterboards process the data. The architecture of the daughterboards is based on programmable components (FPGAs) and Digital Signal Processors, precisely around the TMS320C6414, the last DSP generation from Texas Instrument.

A detailed description of the architecture of the ROD boards is given in this paper, as well as the status of the project and future prospects.

General Terms

Algorithms, Design , Performance.

Keywords

ATLAS, ReadOut Drivers, FPGA, TMS320C6414, DSP.

1. OVERALL PRESENTATION

1.1 The ATLAS experiment

ATLAS is one of the four experiments of the Large Hadron Collider (LHC) currently under construction at the CERN Laboratory in Switzerland. Its goal is to explore the fundamental nature of matter and the basic forces that shape our universe. ATLAS is the largest collaborative effort ever attempted in the physical sciences.

ATLAS will study proton-proton collisions at a center of mass energy of 14 TeV. It consists of various sub-detectors which analyze different aspects of a collision. The Liquid Argon calorimeter [1] is one of the key sub-detector in ATLAS. It is tailored to identify electrons, photons and hadrons, and measure the energy carried by these particles. In total, about 200 000 cells outputs are read from this calorimeter. A high signal sampling frequency (40 MHz), a large energy dynamic range of the readout cells (from 50 MeV up to 3 TeV) and a good energy resolution are some of the main challenges of the Liquid Argon readout electronics.

1.2 The ROD modules in the electronic chain

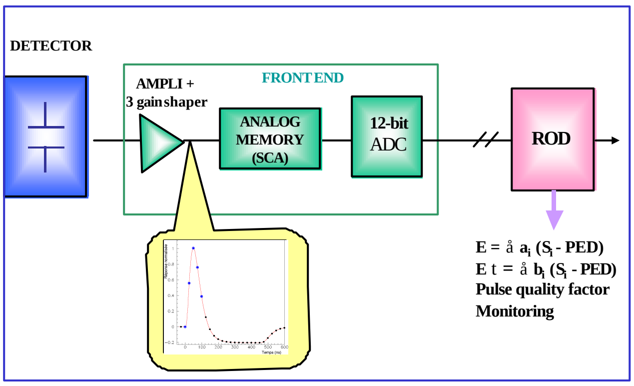

Figure 1 : The upstream electronics chain of the ATLAS Liquid Argon Calorimeter

Charged particles, produced by the Hadron collisions, induce a current in the calorimeter cells. For each cell, the analog signal is treated in the Front End Board (FEB), where it is amplified, shaped, sampled and stored in analog form in a switched capacitor array every 25 ns. Upon receipt of a Level One trigger (signal which selects the interesting events at a maximum rate of 100kHz), five (or more) samples are digitized by a 12-bit ADC and sent on optical links towards the 200 ROD modules [2], where they are processed. Figure 1 shows the upstream electronics chain of the Liquid Argon Calorimeter.

1.3 The ROD modules goals

A single ROD module receives data from 8 FEBs, that is (typically) five digitized samples from 1024 calorimeter cells. Data arrive at the frequency of the LHC (40 MHz).

The module is in charge of calculating the energy and the time relative to the peak of the signal for each channel, along with a pulse quality factor, which indicates how closely the samples follow the expected waveform.

The algorithm implemented in the ROD to extract the energy and time for each channel, uses a technique called optimal filtering [3]. The idea is to estimate these quantities in an accurate and computationally efficient way, minimizing the background noise contributions. The energy (E) and time (T) are expressed as a weighted sum of the samples Si, as shown in the following expressions:

$$E = \sum a_i \,.(S_i - PED)$$
$$E.T = \sum b_i \,.(S_i - PED)$$

where i extends over all samples, PED is the pedestal value, and a_i and b_i are the optimal filtering weights.

The pulse quality factor is a normal chi squared calculation:

$$\chi^2 = \sum ((S_i - PED) - E.\, g_i)^2$$

where g_i is the expected normalized waveform for a given channel.

The error on the energy is amplitude independent, whereas the error on the time varies inversely with the amplitude. For this reason, it only makes sense to calculate T for those channels with E above some threshold value. For a given event, most of the cells have low energy, coming from background noise. There are few cells for which T, and χ^2 must be calculated. Simulations show that this fraction of high energy cells is around 10 %.

Since the raw data from the FEB are no longer available offline, the ROD module must perform monitoring of the calorimeter functioning by building histograms.

During calibration runs, charges of various amplitudes are injected in the electronic chain. The ROD modules compute first and second moments and send data to a local processor, which then calculates calibration constants (a_i, b_i in the formula) for each channel of the calorimeter.

1.4 Requirements

The main requirements for the ROD system are the following:

- High channel density.
- The maximum Level 1 trigger rate for ATLAS is 100 kHz. So, the ROD module must be able to process an event in less than 10 µs, including histograms.
- Use of commercial programmable processor. A natural choice is Digital Signal Processors (DSP), because they present a very efficient calculation power for that kind of algorithm and a high I/O bandwidth.
- Modular design. Basic components should be easily changed/upgraded.
- Low power consumption.

2. The ROD Module

The ROD module is a 9U VME64x board housed in a 9U VME crate with 21 slots. It is in charge of processing the data and transferring the results to the acquisition system through a transition module located at the back of the crate.

As required, a modular design has been chosen to allow for an easy upgrade of the DSP components. It consists of a motherboard [4] and four daughterboards called Processing Units (PU) [5] mounted on top. Figure 2 shows a simplified scheme of the ROD module.

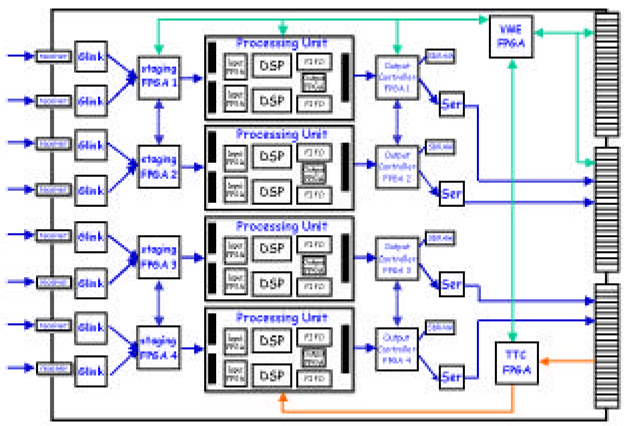

Figure 2 : The ROD module scheme

2.1 The motherboard

Serial data (16 bits @ 80 MHz) are received from each of the eight FEBs by the ROD motherboard through an optical receiver and de-serialized by a Glink chip. Four FPGA chips, called staging FPGAs, route the data from the Glink chips to the PU boards. Two DSPs are mounted on each PU and perform the optimal filtering calculations. The DSP output data are stored in two FIFOs on the PU. Four FPGAs on the ROD motherboard, called Output Controller, get the data from the FIFOs and send them to Synchronous Dynamic Random Access Memory (SDRAM) for monitoring purposes and to the serializer chips for the acquisition sytem. These latter serialize and send the data in LVDS signals at 280 MHz to the transition module.

The VME FPGA interfaces the ROD with the VME bus and deals with the busy signal (signal generated by the ROD to stop the Level One Trigger: for example, in case the DSP is busy with data processing). The TTC FPGA gets and distributes the Trigger Timing and Control information of the experiment.

2.2 Staging mode

At the beginning of LHC, due to contingency, the ROD motherboard will be equipped with only half of the PUs. This is called the staging mode, where the trigger rate will be kept bellow 50 kHz. This is the reason why a data bus between staging FPGAs (32 bits at 80 MHz) has been introduced. Data from four G-link chips are routed through one staging FPGA to one PU board. Therefore, in staging mode the DSP will process twice as many channels as in normal mode with all PUs.

2.3 The Processing Unit

2.3.1 The Processing Unit Architecture

Figure 3 shows the Processing Unit architecture :

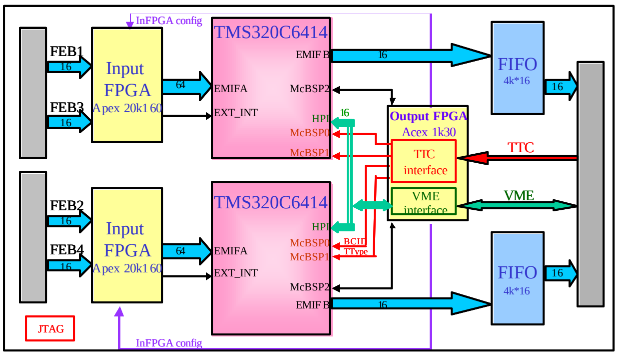

Figure 3 : The Processing Unit block diagram

The Processing Unit is a 120*85 mm board, composed of two DSP blocks, each able to treat up to 128 calorimeter channels (1 FEB) in normal mode and 256 channels (2 FEB) in staging mode. Each DSP block is composed of an input FPGA (InFPGA), a TMS320C6414 DSP from Texas Instrument and a 4k*16 bits deep output FIFO. The TMS320C6414 was chosen for its very high power calculation and its important set of peripherals (see section 2.3.2).

Input FEB data first enter the InFPGA where they are checked and formatted as needed for the DSP algorithm. When an event is ready, an interrupt is sent to the DSP which launches a DMA to read the data on the 64-bits EMIFA bus. Once the DSP has finished to process an event, it writes the results in the output FIFO through the 16 bits EMIFB bus. DSP DMA transfers run at 100 MHz.

The mezzanine contains also an output FPGA used for the TTC and VME interface. It allows, in particular :

- TTC signals transmission to the DSP through 2 serial ports (McBSP).
- PU control from the VME bus.
- DSP boot and histograms read through the 16-bits Host Port Interface (HPI) of the DSP.
- Full duplex serial port (McBSP2) with each DSP (DSP commands, status read).
- InFPGA boot and configuration.

2.3.2 The TMS320C6414 DSP architecture

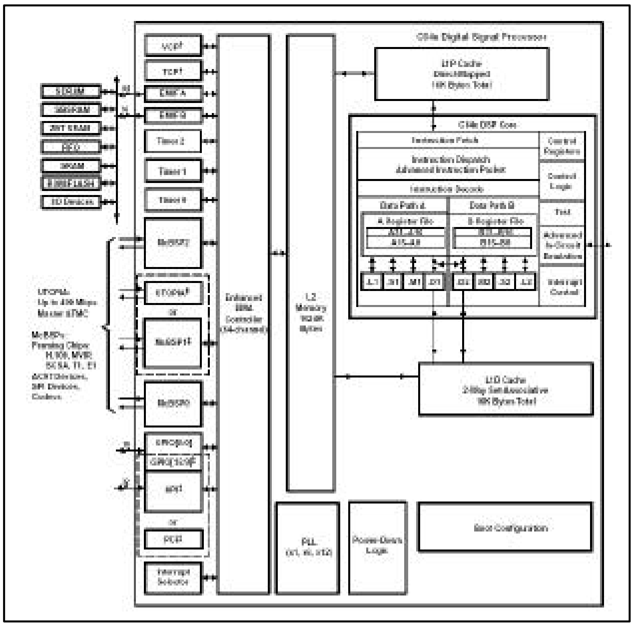

Figure 4 : The TMS320C6414 architecture

The TMS320C6414 is one of the highest performance fixed-point DSP in the Texas Instrument DSP family [5]. The core processor has 64 general-purposes registers of 32-bit word length and eight independent functional units (two multipliers for a 32-bit result and six arithmetic logic units). The core is based on an advanced Very Long Instruction Word (VLIW) architecture, allowing up to eight 32-bit instructions to feed the eight functional units every clock cycle. The DSP clock rate is 600 MHz.

The TMS320C6414 uses a two-level cache based architecture. The first level is a set of 128 kbit of program cache and 128 kbit of data cache. The second level consists in a 8 Mbit memory space.

The TMS320C6414 has also a powerful and diverse set of peripherals, in particular 3 multichannel full duplex buffered serial port (McBSP), a user-configurable 16-bit or 32-bit host port interface (HPI) and two glueless external memory interfaces (64-bit EMIFA and 16-bit EMIFB).

Figure 4 shows the TMS320C6414 architecture.

2.3.3 *Software description*

The code is organized around a specifically designed real time preemptive kernel called RTX (Real-Time Executive). The RTX can handle up to 32 tasks of different priorities and provides the standard inter-task communication services: semaphores, messages, mail-boxes [7] [8].

This kernel, tailored to optimize the ROD specific needs, has also the advantages to be scalable and easy to upgrade.

The RTX requires a small memory space (1.5 kbytes) and adds a small CPU overhead of less than 3%.

Data come in through two circular buffers (16 events deep) and go out also through a circular buffer (16 events deep). These buffers allow for incoming data rate fluctuations. Indeed, rate is 100 kHz on average, but can fluctuate above.

Only a few tasks are required. The first is the synchronization task, which checks for consistency between FEB and TTC data. The second task is the process function. This process is either Physics, Test or Calibration.

After both synchronization and process tasks have completed, the send task is executed. This task prepares data in the output buffer so they can be properly sent out to the motherboard.

Every external transfer is handled by the enhanced DMA controller (EDMA). The EDMA interrupt subroutine (ISR) is woken up every time a DMA transfer is finished (FEB data, TTC data or output). It is used to increment or decrement counters, allowing the DSP to know how many events are stored in each circular buffer. These counters are used for the busy signal generation to control the data flow.

Figure 5 shows the DSP code structure.

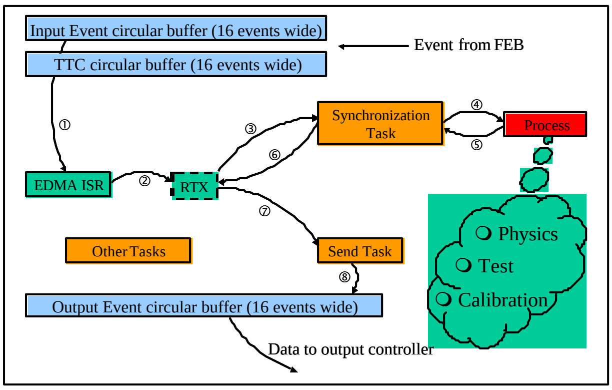

Figure 5 : The DSP code structure

3. STATUS and PROSPECTS

3.1 Status

The ROD module including PUs with 2 DSP blocks, will be ready to be tested by the end of February 2003. However, the architecture of the motherboard and PU has been mostly validated around a single DSP board. Figure 6 shows the single DSP PU prototype.

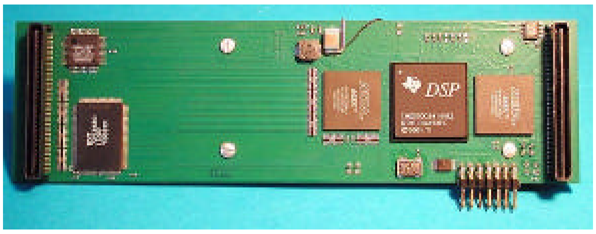

Figure 6 : The single DSP PU prototype.

The DSP software is developed with Code Composer Studio. Few differences were seen between simulator results and actual measurements.

The whole code is written in C language, apart from the physics loops which is coded in linear assembly, and then optimized using the Code Composer Studio program. This presents several advantages : code complexity decreased, better legibility and maintenance.

Simulations show that it takes about 3.5 µs for the physics calculation of 128 channels, including the necessary histograms and a fraction of 10% of high energy cells, for which the time and chi square are calculated.[9]

It is important to underline that about 30 to 40% of this time is due to stall cycles, i.e. cycles lost because instructions or data are not in the L1 cache memory. This causes the CPU to stall for a certain period of time, until the data or instruction is copied into the cache.

The ROD algorithm needs the use of a lot of data and these stall cycles cause the loss of a lot of time. They are one of the main disadvantages of the TMS320C6414 DSP in our project. To face this problem, we tried to optimize the way data are organized in the main memory, so that the number of misses introduced by the cache memory is minimized [10].

If the physics calculations take about 3.5 µs, the complete code execution takes about 7 µs (including the RTX kernel, the synchronization and send tasks, the EDMA ISR management), leaving 30 % of margin for further improvements in the ROD algorithm. This margin is today acceptable by the physicists.

3.2 Prospects

The next steps are the following:

- April 2003 : validation of the new motherboard and the double DSP PU in standalone mode.
- Fall 2003 : System tests in the experiment environment (data coming from FEB and TTC system, outgoing to the data acquisition, …)
- Spring 2004 : production launch (200 ROD modules + spares).
- Summer 2004 : Boards installation at LHC.

4. CONCLUSION

The technical requirements of the Read Out Driver for the liquid argon calorimeters in ATLAS have been described and the architecture of the ROD boards was presented. ROD prototypes show very encouraging results. They demonstrate the absence of blocking issues and respect the ATLAS experiment bandwidth with some comfortable margin. The TMS320C6414 DSP is essential to reach the collaboration objectives.

However, a lot of work has still to be done to validate the prototype in the experiment environment, produce and install all the boards at the LHC.

5. REFERENCES

[1] ATLAS collaboration, ATLAS Liquid Argon Calorimeter Technical Design Report, N/LHCC/96-41ATLAS TDR 2, 15 Dec1996. http://atlas.web.cern.ch/Atlas/GROUPS/LIQARGEXT/TDR/html.1812/Welcome.html

[2] The LARG ROD community, The ROD Demonstrator description document, November 2002 http://wwwlapp.in2p3.fr/~poggioli/Working_version.doc

[3] W.E Cleland, Signal processing considerations for Liquid ionization calorimeters in a high rate environment, NIM A338 (1994) 467-497

[4] A.Blondel et al, The ROD Mother Board for the ATLAS Liquid Argon Calorimeters. Board description. http://www.cern.ch/Imma.Riu/

[5] J.Prast, The ATLAS Liquid Argon Calorimeters ROD, the TMS320C6414 DSP Mezzanine board PU documentation. http://wwwlapp.in2p3.fr/~poggioli/pu6414final.doc/

[6] Texas Instrument, TMS320C64X Fixed Point Signal Processor, October 2002, http://www-s.ti.com/sc/ds/tms320c6414.pdf

[7] Nicolas Chevillot, Global view on hardware and Code Structure, September 2002, http://wwwlapp.in2p3.fr/informatique/Atlas_online/Documents/Dsp/ATL-ROD-RTX/AOS_ATL-TESTDEMO-GEN-1.2.pdf

[8] Gelu Ionescu, Software Requirement Specification for Real Time executive (RTX) on TIC64x, October 2001, http://wwwlapp.in2p3.fr/informatique/Atlas_online/Documents/Dsp/ATL-ROD-RTX/AOS_RTX-TI-SRS1.pdf

[9] http://wwwlapp.in2p3.fr/informatique/Atlas_online/Documents/Dsp/ATL-ROD-RTX/AOS_ATL-DSP-PHYS.pdf http://wwwlapp.in2p3.fr/informatique/Atlas_online/Documents/Dsp/ATL-ROD-RTX/AOS_ATL-DSP-PHYS.pdf

[10] Nicolas Chevillot, Stalls problems and solutions, September 2001, http://wwwlapp.in2p3.fr/Electronique/Experiences/ATLAS-ELEC/Rod/Document/Stalls.pdf