On the developments of the Read Out Driver for the ATLAS Tile Calorimeter.

J. Castelo[1], V. González[2], E. Sanchis[2], J. Torres[2], G. Torralba[2], J. Martos[2]

[1]IFIC, Edificio Institutos de Investigación - Polígono la Coma S/N, Paterna, Valencia, Spain
Jose.Castelo@ific.uv.es
[2]Dept. Electronic Engineering, Univ. Valencia, Avda. Dr. Moliner, 50, Burjassot (Valencia), Spain
Vicente.Gonzalez@uv.es, Enrique.Sanchis@uv.es, Jose.Torres@uv.es, Gloria.Torralba@uv.es, Julio.Martos@uv.es

Abstract

This works describes the present status and future evolution of the Read Out Driver for the ATLAS Tile Calorimeter. The developments currently under execution include the adaptation and test of the LiAr ROD to Tile Cal needs and the design and implementation of the PMC board for algorithm testing at ATLAS rates.

The adaptation includes a new transition module with 4 SLINK inputs and one output which match the initial TileCal segmentation for RODs. We also describe the work going on in the design of a DSP-based PMC with SLINK input for real time data processing to be used as a test environment for optimal filtering.

I. INTRODUCTION

At the European Laboratory for Particle Physics (CERN) in Geneva, a new particle accelerator, the *Large Hadron Collider (LHC)* is presently being constructed. In the year 2006 beams of protons are expected to collide at a center of mass energy of 14 TeV. In parallel to the accelerator, two general purpose detectors, *ATLAS* and *CMS*, are being developed to investigate proton-proton collisions in the new energy domain and to study fundamental questions of particle physics .

This new generation of detectors requires highly hardened electronics, able to deal with a huge amount of data in real or almost real time. The work we present here is included in the studies and development currently carried out at the University of Valencia for the Read Out Module (ROD) of the hadronic calorimeter TileCal of ATLAS.

II. THE TILECAL ROD SYSTEM

TileCal is the hadronic calorimeter of the ATLAS experiments. It consists, electronically speaking, of 10000 channel to be read each 25 ns. Data gathered from these channels are digitized and transmitted to the data acquisition system (DAQ) following the assertions of a three level trigger system [1].

In the acquisition chain, place is left for a module which has to perform preprocessing and gathering on data coming out after a good first level trigger before sending them to the second level. This module is called the Read Out Module (ROD).

For TileCal, the ROD system will be built most probably around custom VME boards which will have to treat around 2 Gbytes/s of data. Intelligence will be provided to do some preprocessing on data.

For the reading of the channels we are working on a baseline of 64 ROD modules. Each one will process more than 300 channels. The studies currently going on at Valencia focus on the adaptation of the first prototype of the LiArg ROD to TileCal needs.

The basic schema to use is based on the ROD crate concept in which ROD modules are grouped into VME crates jointly with a Trigger and Busy Module (TBM) and possibly other custom cards when needed. This ROD crate interfaces with the TileCal Run Control and the ATLAS DAQ Run control. Figure 1 shows this structure schematically [5].

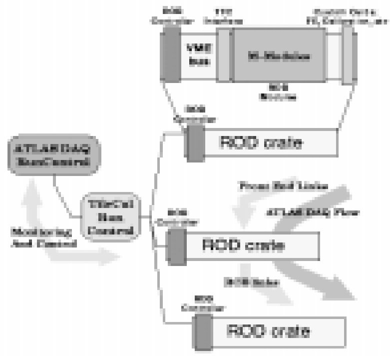

Figure 1: TileCal ROD System

The basic functions and requirements of all ATLAS ROD can be found in [1] and may be summarize saying that the ROD board receives the data from the FEB which after some processing are sent to the ROB. These data may be buffered to be able to work with the maximum LVL1 trigger rate (100 KHz) without introducing extra dead time.

For each particular detector, some preprocessing could be done at ROD level. For TileCal, RODs will calculate energy and time for each cell using optimal filtering algorithms besides evaluating a quality flag for the pulse shape (χ^2). RODs will also do the data monitoring during physics runs and make a first pass in the analysis of the calibration data leaving the complete analysis to the local CPU of the ROD crate.

In some cases data will not be processed and will flow raw to LVL2. These include interesting events, large energy depositions in a cell or debugging stages.

It will be also desirable to have the functionality to apply corrections to the energy or time estimators for example to correct the non-linearities in the shaper or in the ADC.

Finally ROD will monitor the Minimum Bias and pile-up noise and will have the possibility of working in special runs at reduced trigger rate.

III. LIARG ROD AND TILECAL ROD

As mentioned before, our work develops now in the direction of adapting the first LiArg ROD prototype [7] to TileCal needs exposed before. The main reason to do this is the great similarity of the two detectors and the great difference in the requirements which make LiArg solutions suitable, with modifications, for TileCal.

The basic differences in the ROD concept for LiArg and TileCal rise, in a first approximation, in the working baseline which is summarized in table 1.

Table 1: ROD baseline for LiArg and TileCal

	ROD LiArg	**ROD TileCal Baseline**
Input links (32 bits @ 40 MHz	2	4
Number of channels per board	256	154 (2*64b+2*31eb)
Number of DSP Processing Units	4	4
Number of channels/DSP PU	64	46b or 31eb
Output Links (800 Mb/s)	1	1 (1,14 Gb/s, expected)

The block diagram of the LiArg ROD prototype is shown in figure 3. It is based in a 9U VME motherboard which holds four DSP-based processing units (PU) as mezzanines. These mezzanines are based on TI C6202 DSPs at 250 MHz with some external logic: FIFOs, FPGAs and memory. Figure 2 shows the block diagram of the PUs [8].

The input and output of data is place on a 9U transition module. For the first prototype this module has only two inputs and one output.

To adapt this solution to TileCal needs we need to reconsider the following aspects:

- Data input/output format and rates: we need 4 inputs and one output at the transition module. This implies a new design of this transition board.
- Processing power: because of its great number of channels, LiArg DSP PU have a lot more computing power than needed in TileCal. This issues the question of whether it is necessary to use exactly the same type of PUs or we could use cheaper ones even not based on DSPs but on FPGAs.

Because of the modularity of the LiArg solution, our work focuses on the design of a new transition module, leaving the decision about the PU postponed.

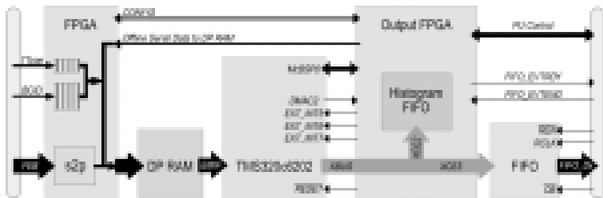

Figure 2: Block diagram of the DSP Processing Units.

IV. THE TM4PLUS1 TRANSITION MODULE

Lets now get into the description of the new design carried out to adapt TileCal inputs to the LiArg motherboard. This new transition module is called TM4Plus1.

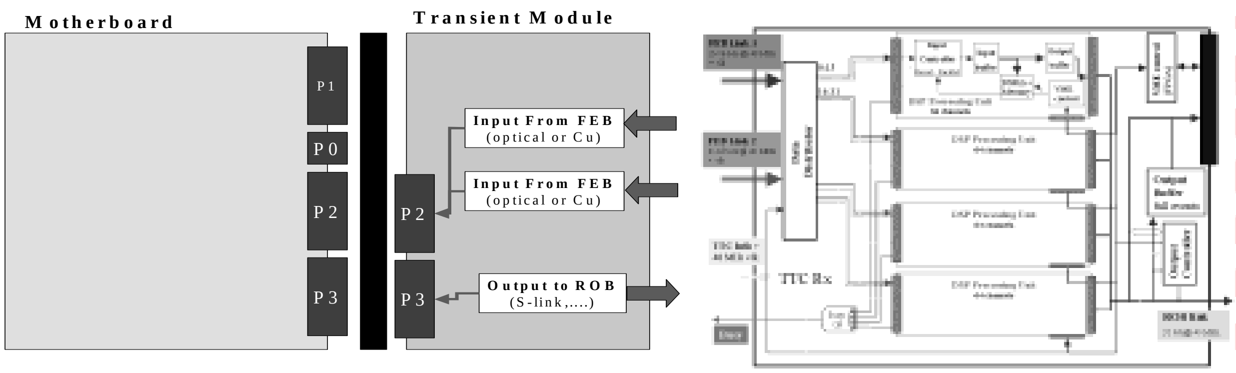

Figure 3: LiArg ROD prototipe

This module has been developed and implemented at CERN by the EP/ATE group. Its block diagram is shown in figure 4.

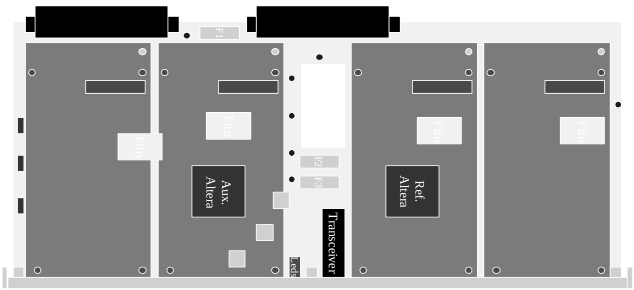

Figure 4: TM4Plus1 block diagram.

The transition module is a modified version of the one used by LiArg that includes 4 input SLINK channels in PMC format and 1 GLINK output integrated in the PCB. The PMC input channels are capable of reading 4x32 bits at 40 MHz and allow us to test different input technologies. The output will also run at 40 MHz with a data width of 32 bits [4] [6].

On the board there are also 4 input FIFOs, 4Kwords each, to accommodate the differences between input speed and processing on the FPGAs.

These lasts are implemented on two ALTERA devices. The tasks of each of the FPGAs are:

- Reformatting Altera: data multiplexing and SLINK control. This devices will reformat and merge data in a 4 to 2 manner to produce data similar to what the motherboard is expecting if used with LiArg detector.
- Auxiliary Altera: it holds the code for the integrated ODIN output. The free space left will be used in conjunction with the reformatting Altera.

The data flow to these FPGAs is shown in figure 5. Following the tasks division the reformatting Altera receives the data from the 4 input channels to perform the 4 to 2 multiplexing. Data is sent to the motherboard through P2 connector on the VME backplane with the same format as LiArg.

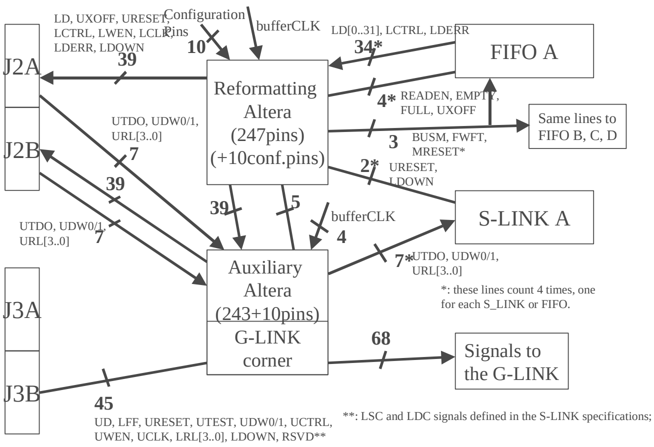

Figure 5: Data flow on the TM4Plus1.

The other Altera receives some lines from the first and hold the ODIN output code. Data once processed at the motherboard is sent through P3 connector to this FPGA to be sent to ROBs.

The basic data processing on the reformatting Altera relates to the forma conversion between TileCal and LiArg. Fortunately, front-end data format on both detectors is quite similar and we have to deal only with a data width problem. This problem is due to the fact that on the motherboard, the 32-bit path of P2 connector is splitted into two 16-bit paths each one going to a PU. For TileCal with have 32-bit paths already in the front-end data, so we have to divide these data into 16-bit blocks and send them consecutively to the motherboard.

There is also a problem with data control due to the way the LiArg motherboard controls the data flow. Situations may arise when we could have data only on one of the two input provided to first check if this occurs and second to solve it.

Our proposal implements a time-out activated on the arrival of the first data on one of the input links to be multiplexed. If after the time out no data are received, the space for that channel is filled with zeros and a flag is set on the header to let the PU treat these data as no-data instead of zeroes.

These two processes are depicted schematically in figure 6.

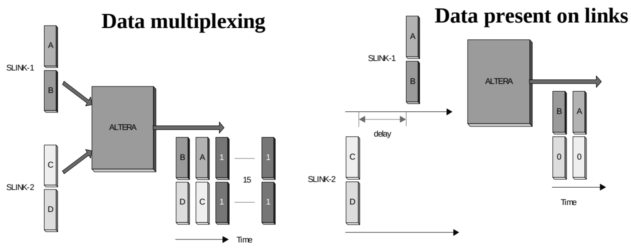

Figure 6: Data multiplexing using FPGAs

V. Present developments

At IFIC and University of Valencia there are three development fronts undergoing.

The main is related with the tests and developments for the TM4Plus1 board and the final ROD prototype, the second goes towards the design and implementation of a PMC card for algorithm tests and the third deals with the software issues at the ROD controller.

Lets review now the current status on each of these directions.

A. *The TM4Plus1 board and final ROD prototype*

The tasks on the TM4Plus1 board currently going on are of two kinds:

Hardware:

•Test of motherboard and PUs. This is already finished.

•Test of the TM4Plus1 transition module: not yet finished.

•Design of a custom FPGA based PU: starting.

Software:

•TM4Plus1 FPGAs: going on. This work refers to the implementation of the processing commented before on the two Alteras

•PUs: we need to reprogram the DSPs to do optimal filtering and also the input FPGA.

For the final ROD prototype we are currently designing a new PU based on FPGA instead of DSPs. These will imply a reduction in cost, because the DSP PU are the most expensive component in the LiArg ROD, and an increase on parallelism as we will not be limited by the DSP architecture but by the FPGA capacity. Our working block diagram for a final TileCal ROD prototype is shown in next figure.

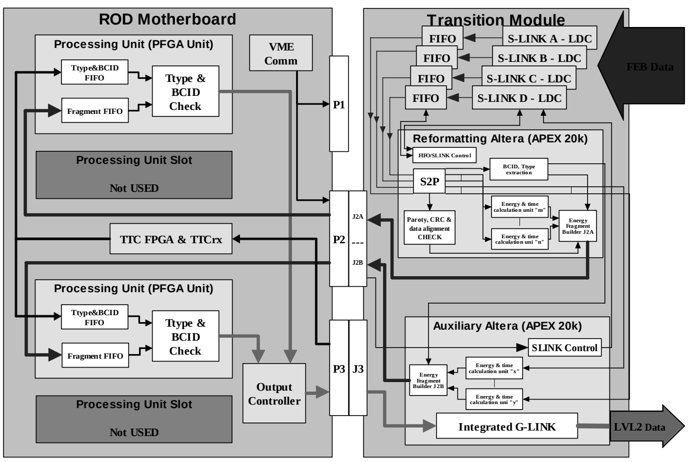

Figure 7: Final TileCal ROD prototype.

As it can be seen we keep the LiArg motherboard to have VME access and the TM4Plus1 board, but substitute the DSP PUs by new FPGA based ones. Also a redistribution of tasks occurs, placing almost all processing issues on the FPGAs of the transition module where energy an timing estimation will take place. On the motherboard only data integrity checking and TTC operation will be done.

By processing data on the transition module we reduce data volume flowing to the motherboard. This opens the possibility of increasing the number of input channels on the transition module by previously integrating them on the PCB (no more PMCs).

B. The SLink PMC card

Parallel to these activities we are also involved in the design and development of the DSP based PMC card with SLINK input for testing the optimal filtering algorithms on a commercial VME processor.

The basic idea is to have a PMC with SLINK input capability and with some intelligence deployed on a FPGA and a TI 6X DSP [2]. We are currently working on the following block diagram.

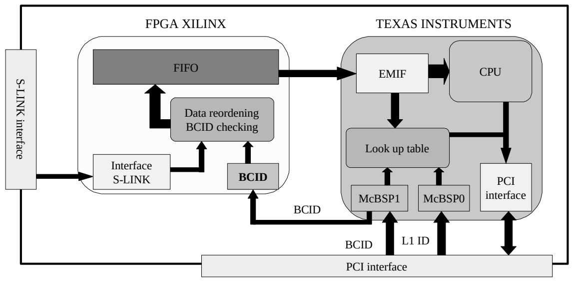

Figure 8: PMC block diagram.

For the DSP we are currently designing for the TMS320C6205 which includes a PCI interface that save us the task of implement this interface on a FPGA. For the FPGA we are designing with XILINX X2CS100 device.

The DSP will load TTC data (BCID, EventID and Trigger Type) using two serial channels to make the data integrity operation and output data formatting, while the FPGA will take care of the SLINK interface, data reordering, BCID sequence check and the EMIF communication with the DSP.

C. The ROD Controller

Activity in this field is focuses on the adaptation of the LiArg ROD software libraries to the setup at Valencia based on a BIT3 VME-PC interface as ROD controller and the TileCal ROD integration with DAQ-1.

The adaptation of the LiArg libraries is already finished and has required some effort on the driver side. For the DAQ-1 integration the work foreseen is the development of the Local ROD VME software, the online software and ROS dataflow. We expect to start this work very soon.

VI. REFERENCES

[1] ATLAS Trigger and DAQ steering group, “Trigger and Daq Interfaces with FE systems: Requirement document. Version 2.0”, DAQ-NO-103, 1998.

[2] TEXAS INSTRUMENTS, “TMS320C6205 Data sheet,” Application Report SPRS106, October 1999

[3] K. Castille, "TMS320C6000 EMIF to external FIFO interface," Application Report SPRA543, May 1999

[4] J. Dowell, M. Pearce “ATLAS front-end read-out link requirements,” ATLAS internal note, ATLAS-ELEC---1, July 1998

[5] C. Bee, O. Boyle, D. Francis, L. Mapelli, R. MacLaren, G. Mornacchi, J. Petersen, “The event formatr in the ATLAS DAQ/EF prototype-1,” Note number 050, version 1.5, October 1998

[6] O. Boyle, R. McLaren, E. van der Bij, “The S-LINK interface specification,” ECP division CERN, March 1997

[7] The LArgon ROD working group, “The ROD Demonstrator Board for the LArgon Calorimeter”

[8] S. Böttcher, J. Parsons, S. Simion, W. Sippach “*The DSP 6202 processor board for ATLAS calorimeter*”