ROD General Requirements and Present Hardware Solution for the ATLAS Tile Calorimeter.

J. Torres[1], J. Castelo[2], E. Fullana[2]

[1]Dept. Electronic Engineering, Univ. Valencia, Avda. Dr. Moliner, 50, Burjassot (Valencia), Spain
Jose.Torres@uv.es
[2]IFIC, Edificio Institutos de Investigación - Polígono la Coma S/N, Paterna (Valencia), Spain
Jose.Castelo@ific.uv.es, Esteban.Fullana@ific.uv.es

Abstract

This work describes the general requirements and present hardware solution 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 TileCal needs. In fact, our actual work is centred in the new ROD Motherboard design and to program the Staging FPGA for TileCal and LiAr.

Nevertheless, there are more activities under development such as software studies of the algorithm for processing the detector data (Optimal Filtering), the implementation of it in Digital Signal Processors and the integration of the system in the DAQ chain using Online Software (DAQ-1 project).

I. INTRODUCTION

At CERN in Geneva, a new particle accelerator called Large Hadron Collider (LHC) is being constructed. It will reach energies of 14 TeV and it will be finished in 2007. Four experiments (ATLAS, CMS, ALICE and LHCb) are being developed in order to investigate and explore all the possibilities of LHC.

The work we present here is included in the studies and development currently carried out at the University of Valencia and IFIC for the read Out Driver (ROD) of the hadronic calorimeter (TileCal) of ATLAS.

II. THE TILECAL ROD SYSTEM

TileCal ROD is the hadronic calorimeter of the ATLAS experiments. The TileCal ROD has to read and compute 9856 channels each 10µs and it must be able to works in real time. The data gathered from these channels are digitized and transmitted to RODs with high-speed optical links.

Each ROD module must have the ability to process this data, and send it through an output optical link to the next stage (ROBs) in the data acquisition chain. Also, the ROD system must provide some communication for monitoring and control of all the RODs modules, and this feature is driven by the ROD controller (SBC), which is the master CPU of the ROD crate for controlling the ROD modules (slave devices).

There are another slave module that must be built for this application, and it's the TBM, which is responsible for receipt the TTC information at ROD crate level and distribute it to all ROD modules, by other hand it has another function that is to recollect and implement an OR function over all the BUSY signals of the ROD modules in order to stop the L1A generation. A bidirectional communication with CTP is done through a TTC crate in the partition, managing BUSY and TTC signals.

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.

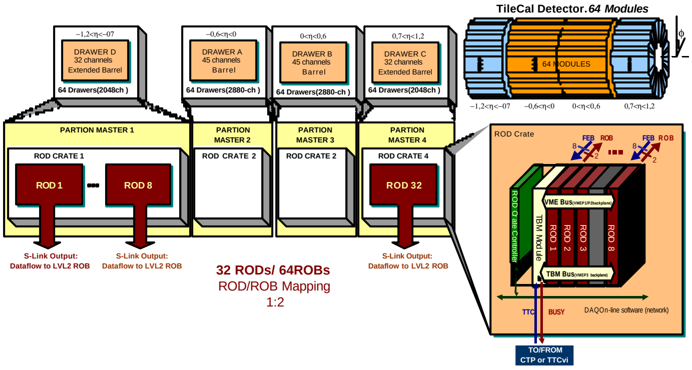

Figure 1: TileCal ROD System

III. LIAR ROD AND TILECAL ROD

The motivation of this part is the comparisons of the TileCal ROD requirements with the LiAr new motherboard design for take advantage of the new more integrated board due to cost reduction. In this section, a comparison between LiAr requirements and TileCal should report a final hardware solution for the TileCal ROD dataflow.

In Table 1 is summarized the read out driver performance requirements for the hadronic calorimeter.

Number of Channels	**9856**
Number of Drawers (FEB)	**256**
Number of channels per drawer (EB)	**32**
Number of channels per drawer (CB)	**45**
Number of Drawers (EB)	**128**
Number of Drawers (CB)	**128**
Input event size per FEB (7 samples) in kbytes [2]	**0,57**
Total input event size (7 samples)	**147,00**
Input Data Bandwidth @ 100kHz Lvl1 ATLAS rate Gbytes/sec	**14,02**
Number of Drawers (FEB) per ROD	**4**
Number of RODs	**64**
Typical output event size per ROD (Typical Size 1) in kbytes [2]	**1,10**
Output Data Bandwith @ 100kHz Lvl1 ATLAS rate Gbytes/sec	**6,70**
Number of PUs per ROD	**4**
Number of PU (DSP) instructions per channel (seven samples). Aplying Optimal filtering (E, t and chi2)	**70**
Total processing power in MIPs	**68992**

Table 1: ROD baseline for TileCal

The block diagram of the LiArg ROD prototype is shown in figure 2. 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 3 shows the block diagram of the PUs.

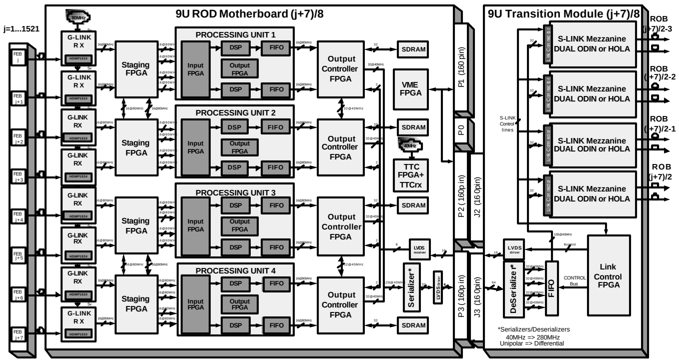

Figure 2: LiAr ROD Motherboard.

LiAr needs more processing power per link, 128 channels/link (LiAr), 45 for CB and 32 for EB ch/link (TileCal); so only 2 Processing Units, and 2 Output Controllers plus SDRAM data storage are enough for Tiles dataflow needs.

Input bandwidth: The maximum input BW of each link for a TileCal physic event is 467.2 Mbit/sec, so 4 links (4 drawers) is 1,825 Gbits/sec. Input bandwidth of the Processing Unit is 2,5Gbits/sec (64bits@40Mhz) => One PU has enough input BW for 4 links.

The processing unit: We need to process 154 channels (four drawers) in two TMS320C6414@600MHz DSPs (4800 MIPS each). This DSP has the same core with some improvements in number of registers and an enhanced DMA unit over the actual DSP we have tested is the TMS320C6202@250MHz (2000 MIPS). Our actual lab routines could process 45 channels in around 5.5ms (assembler) or 15.5ms (C code).

Potentially, we could process 154 channels with the new PU TMS320C6414@600MHz with 9600 MIPs (two DSPs) in 3.92 ms (assembler) or around 11ms (C language). Because our limit is 10ms at LVL1 100KHz rate, thus if we believe in improvements in the C compiler from Texas Instruments, probably we could program the final system in a better maintainable C code and only with 2 Processing Unit mezzanines installed in the motherboard.

Output Bandwidth: The typical BW for 154 channels (four drawers) is 656 Mbits/sec. Then, an Output Controller FPGA of 1.28 Gbits/sec (32@40MHz) has enough BW for the output of each Processing Unit (154 channels each).

Transition Module: 2 mezzanine links are enough for this configuration.

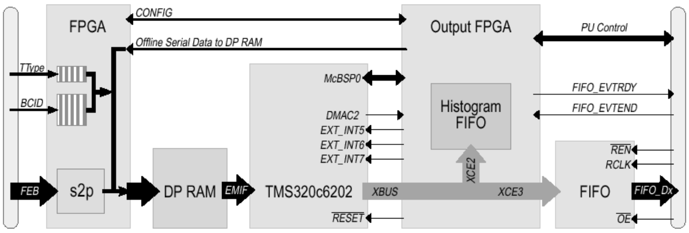

Figure 3: Block diagram of the DSP Processing Units.

IV. LiAr MotherBoard Adapted to TiCal

In Figure 4 is shown the TileCal ideal solution in case of redesigning the input stage of this board. The TileCal front end interface links are implemented with G-LINKs 3,3v HDMP1032 chips running at 40MHz, and a TX system of two fibers which send the same fragment for data error check in the ROD reception side.

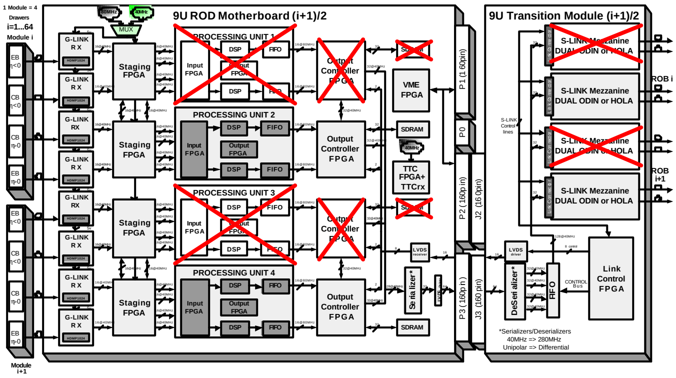

Figure 4: TileCal ROD Motherboard.

The changes proposed are to put double optical receiver and two 3.3v HDMP1034 RX chips with 40 MHz clock for data de-serialization and staging FPGA machine clock. A FIFO for at least two input events is suggested because we need a temporary storage for the redundancy link data, while we are checking the CRC for one of them, in case of errors we read from the FIFO the other link fragment and we use this one if it's OK. If not an error flag must be reported.

The rest of the board is maintained as the original design but without using two Processing Units, and two Output Controllers plus SDRAM data storage. Of course, the motivation of this is for saving costs.

With this solution we reduce the number of RODs from 64 two 32 maintaining the readout in 64 links to ROBs. Therefore we don't spoil the granularity of the read out system.

Is possible to receive the two redundant fibers, and we use the HDMP1034 chips at 3.3v with no cooling problems, and can select between using or not the enhanced simplex mode (pin ESMPXENB=0) in the HDMP1032 transmitter interface link.

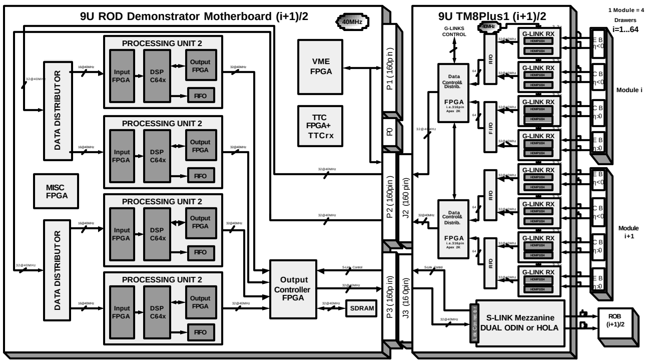

Figure 5: ROD Data Flow.

Hardware design. The routing for the new PCB must be done specifically for TileCal.

The PCB of TileCal and the LiAr one will be different, thus a common order could not be done with less advantages for decreasing costs because of a high number of units request.

The Processing Units are responsible for the data and TTC reception, the implementation of optimal filtering algorithms, the local histogram and to send the processed data to the Output Controller FPGA. A FIFO is needed for having some buffering in the output. It's planned that this Processing unit contains two fixed point Texas Instruments DSP TMS320C6414 running at 600MHz clock rate.

The Output Controller FPGA has to treat the processed data and decide (after previous VME configuration) when to send the events to the output links or to the SDRAM to be available to read through VME bus (low speed data taking). Another possibility is to send to both, in case of spying data for test purposes (≈5% of the data is spying with VME crate controller).

A serializer/deserializer module is implemented because there were not enough pins in P3 backplane to send data for the 4 OC. The trick used is converting the unipolar signals to high noise immunity differential signals like LVDS standard.

This huge amount of data is sent to the ROB through up to four mezzanines S-LINK LSC controlled by an FPGA in the transition module.

The reception of TTC info is done by the TTCrx chip and a TTC_FPGA, which distributes this TTC info to each PU (for compare this TTC info with the one received from the front end data stream) and to the Output Controllers for building the data fragments with the DAQ-1 event format.

The VME FPGA, obviously, is used for the communication with the ROD controller SBC. It is responsible for booting the PU (DSP and FPGA code), and read/write the CSR of Staging FPGAs, OC, TTC FPGA, and in general for access to the control and monitoring of the board.

In the next lines is compared and discussed several options that could be taken around the new design. We will compare three options for the TileCal ROD dataflow architecture, having in minded the advantages and disadvantages of each one.

V. ACTUAL DEVELOPMENTS

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

A. Staging FPGA in ROD Motherboard

The main tasks for the Staging FPGA are:

- It multiplexes the data from the different FEB inputs and sends it to the connector of the PU concerned, depending if it is staging or not. This feature provides the possibility to use only two processing units instead of four, routing the data to the right PU (the staging is configured through VMEbus).
- The Glink chips might need a configuration that will be performed by the staging chips.
- It gets the temperature of the Glink and transmits it to the VME chip. Because these chip usually has high power consumption.
- It transmits the Glink errors (parity and ready) to the PUs and to the VME.

During the tests it will:

- Read the Glink data and transfer it to the VME.
- Transfer data from the VME to the PU. Similar function as 'data distributor block' in the demonstrator board.
- Transfer at high rate some pattern data to the PU.

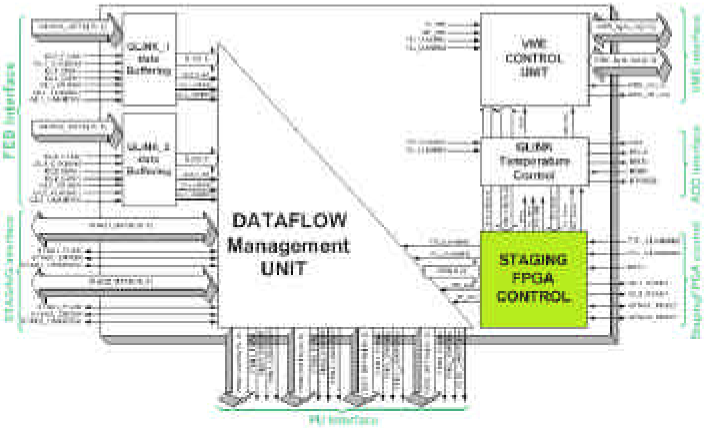

Figure 7: Staging FPGA Internal Block.

B. *Optimal Filtering Implemented in DSP. Processing Unit 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.

About the optimal filtering algorithm, we are using a multisampled method firstly developed for liquid ionization calorimeters. Allows the reconstruction of Energy and time information.

Additionally minimizes the noise coming from thermal sources (electronics) and also from minimum bias events.

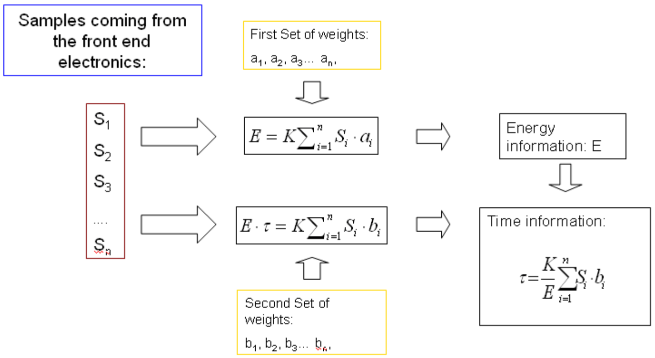
Samples coming from the front end electronics:
S_1
S_2
S_3
....
S_n
First Set of weights:
$a_1, a_2, a_3 \dots a_n,$
$E = K\sum_{i=1}^{n} S_i \cdot a_i$
Energy information: E
$E \cdot \tau = K\sum_{i=1}^{n} S_i \cdot b_i$
Second Set of weights:
$b_1, b_2, b_3 \dots b_n,$
Time information:
$\tau = \frac{K}{E}\sum_{i=1}^{n} S_i \cdot b_i$

Figure 8: Optimal Filtering Block Diagram.

We use optimal filtering for obtain Energy, Tau, and c2. The implementation is considering 7 samples of 10 bits.

Actual studies demonstrate that the resolution will not be improved with different weights for each cell. Use of the same calibration constants table for all channels (this could be changed).

The calculations are 32bit integer except for Multiplication (16bits). Always trying to get the maximum resolution of the integer ALU operations.

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] O. Boyle, R. McLaren, E. van der Bij, "The SLINK interface specification," ECP division CERN, March 1997

[3] The LArgon ROD working group, "The ROD Demonstrator Board for the LArgon Calorimeter"

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

[5] RD12 - Timing, Trigger and Control (TTC) Systems for LHC Detectors. Reference: http://ttc.web.cern.ch/TTC/intro.html

[6] The I/O Dataformat for the TileCal Readout System (J. Castelo) Reference: http://ific.uv.es/tical/rod/doc/rod_data_format%20proposal.pdf

[7] TileCal ROD HW Requirements and LArg compatibility (J. Castelo). Reference: http://ific.uv.es/tical/rod/doc/ROD_tical_HW.pdf

[8] ROD Processing Unit Performance (J. Castelo). Reference: http://documents.cern.ch/archive/electronic/other/agenda/a02281/a02281s2t8/ROD_DSP_performance.pdf

[9] ROD Algorithm Performances using the DSP TMS320C64x. Reference:http://documents.cern.ch/cgibin/setlink?base=atlnot&categ=Note&id=larg-2001-020

[10] Timing, Trigger and Control, and Dead-time handling. Author: Ph. Farthouat . Reference: http://mclaren.home.cern.ch/mclaren/atlas/conferences/ROD/ttc.pdf

[11] TMS320C6202, Fixed-Point Digital Signal Processor. Reference: http://focus.ti.com/docs/prod/productfolder.jhtml?genericPartNumber=TMS320C6202

[12] TMS320C6414, Fixed-Point Digital Signal Processor. Reference: http://focus.ti.com/docs/prod/productfolder.jhtml?genericPartNumber=TMS320C6414

[13] Use of the Central Trigger Processor (CTP) and of the Timing, Trigger& Control System (TTC) for Timing and Calibration (R. Spiwoks) Reference: http://press.web.cern.ch/Atlas/GROUPS/DAQTRIG/LEVEL1/ctpttc/meet_tdaq_121101.pdf

[14] Timing receiver ASIC (TTCrx) Reference Manual. J. Christiansen, A. Marchioro, P. Moreira, and T. Toifl. Reference: http://ttc.web.cern.ch/TTC/TTCrx_manual3.5.pdf