LTPIM – Local Trigger Processor Interface Module

D. Caracinha [a], P. Farthouat [b] , P. Gallno[a] , J. Augusto[a] , G. Evans[a] ,A.Amorim [a]

[a] FCUL, Lisbon University
[b] CERN, 1211 Geneva 23, Switzerland

Abstract

The *Local Trigger Processor Interface Module (LTPIM)* is 6U VME 64x board housing two Altera Cyclone *FPGAs* for VME interfacing and control. It is the interface between the *Central Trigger Processor (CTP)* and the *Local Trigger Processors (LTPs).*

I. INTRODUCTION

The interface between the *ATLAS Central Trigger Processor (CTP)* and the sub-detectors read-out systems is done through the *Local Trigger Processor (LTP)* modules. These modules allow each sub-system to either run in global mode, when it gets the timing and trigger signals from the *CTP*, or in local mode when it handles locally those signals. During the commissioning phase of the detector, and for test purposes, it may be necessary to have several sub-detectors able to run locally with the same timing and trigger signals (e.g the calorimeters and the level-1 calorimeter). Although feasible with the current *LTP* modules, an extra interface module, *LTPIM*, has been designed in order to avoid the need for ad-hoc cabling.

II. LOCATION ON THE *ATLAS*

The *LTPIM* is the interface between the *CTP* and the *LTPs*. There is one *LTPIM* for each *Trigger Timing and Control (TTC)*.

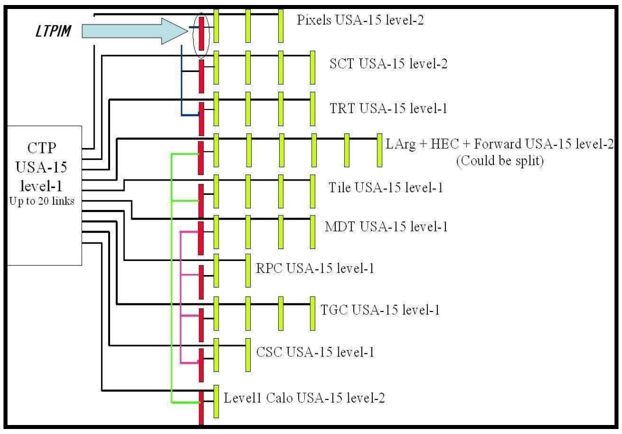

Figure 1: Location on the *ATLAS*

III. *LTPIM* DESCRIPTION

The Board has 8 layers which are two grounds (Analogue and Digital) and has 5 power supplies. The +5v and +3.3v are taken from the *VME* interface and there are DC/DC converters in order to provide for +1.5v, +2.5V and -5V.

There are two *CTP* links for interfacing with the *CTP* and the *LTPs* and two differential ports that connects the Interface modules.

There is also *LEMO* connectors on the front panel that can be used for monitoring signals.

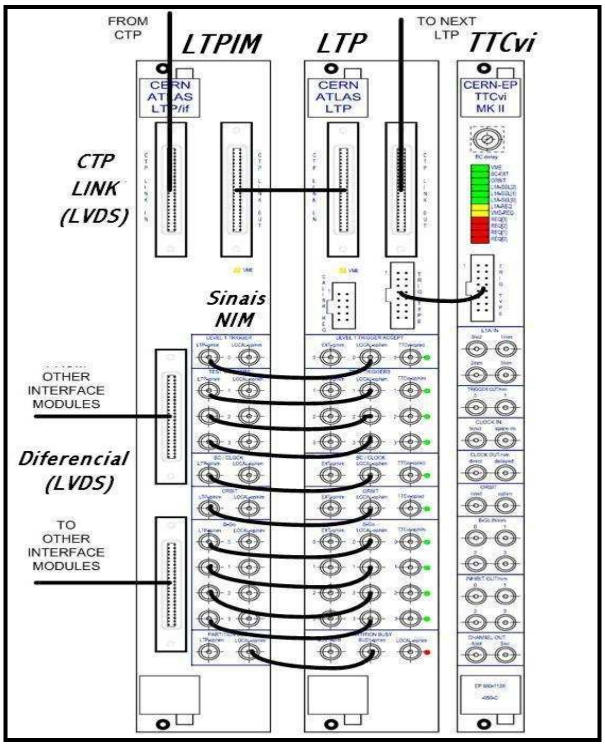

Figure 2: *LTPIM* Front panel

Since the *LTPIM* will be far from the *CTP*, special care has been taken in order to prevent for the possible loss of signal in cables. There are active equalizers for each signal coming from the *CTP*.

These active equalizers can provide a gain and equalization controlled by an analogue voltage. The analogue voltage is provided by a four independent channel *DAC* connected to the control *FPGA*.

There are also delay circuits that are able to delay all the signals (except the *BC* clock) in order to synchronize or to provide necessary delay.

The delay circuits are programmed using an *I2C* bus.

The *I2C* bus master is implemented on the control *FPGA*.

LVDS signals from the *CTP* links and from the differential ports are converted to *LVTTL* for monitoring by the control *FPGA*. *NIM* signals from the *LEMO* inputs are converted to *ECL* and then into *LVTTL*.

All signals can be monitored by *VME* using the monitoring registers implemented on the control *FPGA*.

IV. OPERATION MODE

All connection where made in order to maximize testing capability and according to the *ATLAS* needs.

At the moment it is expected the *LTPIM* to work in three different modes of operation. However the possibility of having other configurations was accounted for. There are the “transparent” mode, the “slave” mode and the “master” mode.

Master and slave mode are to be used mostly during tests and commissioning and transparent mode should be used when the ATLAS detector is running.

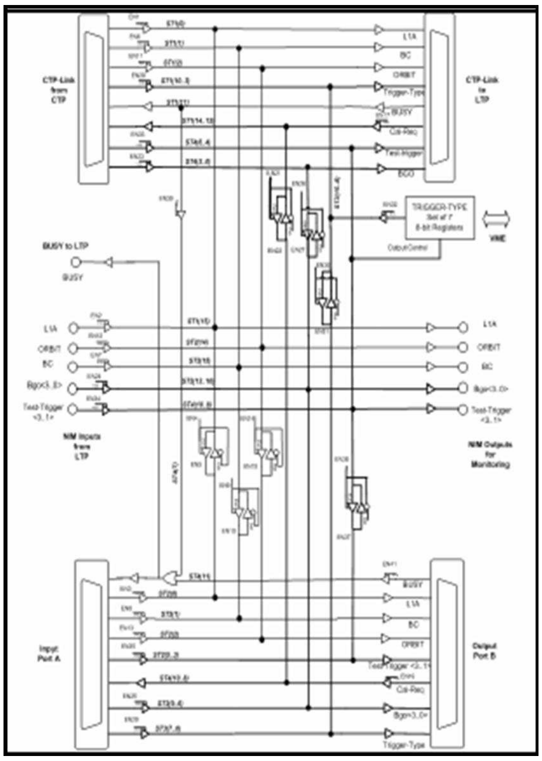

Figure 3: *LTPIM* connections

A. *“Transparent mode”*

All signals from the *CTP* link input are directly sent to the *CTP* link output. Is this configuration the *LTPIM* is only used to provide gain and equalization into the active equalizers on the *CTP* link input and to provide for signal delay on the *CTP* link output. The path delay without any signal delay was kept to minimal and should be 10 -15 ns.

.

B. *“Master mode”*

Using this configuration we have a master *LTP* sending the signals to the *LTPIM*. The master *LTPIM* will send them to the slave *LTPIM* of the slave *LTP* on that partition.

Signals can be put in using the differential input, the *NIM* inputs or the *VME* interface.

The active equalizers may not be used in this configuration but the delay circuits can be quite helpful in testing, calibration and commissioning.

C. *“Slave mode”*

Using this configuration, the slave *LTPIM* receives data from a *LTPIM* master or from another slave and sends it into the *LTP* of the sub partition.

Active equalizers and delay circuits can be used in this configuration.

V. CONFIGURABLE LOGIC

The *LTPIM* houses to Altera Cyclone *EP1C4F324C6 FPGAs* for *VME* interfacing and control. It has an *EPSC4* serial memory configuration device for *FPGA* configuration.
The control *FPGA* is also responsible for the *I2C* bus communication.

A. *“VME Interfacing”*

The *LTPIM VME* interface implements a standard *VME* 64x slave with a simple state machine.

It recognizes address only cycles, data transmission cycles and interrupt cycles. Interrupt cycles are disabled at this moment because it is not expected the *LTPIM* to generate interruptions. However the connections were made in order to account for that possibility in the future.

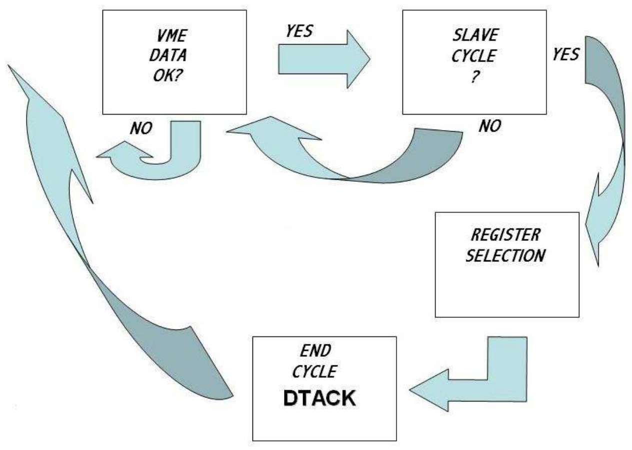

Figure 4: *LTPIM VME* slave simplified schematic

The *LTPIM VME* slave interface is designed for 24 bits addressing and 16 bits data cycle.

Bits 23 to 8 are used to select the *LTPIM* board and bits 7 to 0 are used to select the internal registers. Upon an addressing to the *LTPIM*, information about the register being addressed is sent to the control *FPGA*.

It is generated a delay in order to permit the control *FPGA* to capture data from the *VME* data bus.

When the transmission is complete, the *VME* slave interface generates a *DTACK*, signalling the end of the transmission.

B. *"Control FPGA"*

The control *FPGA* is divided in three blocks: *VME* and monitoring registers, control and tri state gate selection and *I2C* master.

Is has eleven 16 bit registers. Seven of that are accessed by *VME*, the other four are for monitoring.

Register selection is made using a select bit coming from the *VME* interface *FPGA* that will select the proper internal register.

Monitoring registers are used to monitor each line of the *LTPIM*. This condition can be extremely helpful in testing and debugging not only the *LTPIM* but also the *LTPs* and the signals between the *ATLAS* partitions.

Special care has been taken in order to prevent bus conflicts when using the tri state gates. There are codes of operation programmed by *VME* that prevent possible multiple buffers to drive the same bus line.

On power up, the *CTP* links and the differential links are transparent and the *NIM* and transceivers are tri stated.

This is necessary to insure that the board will not be damaged when a cable is connected to his input gates.

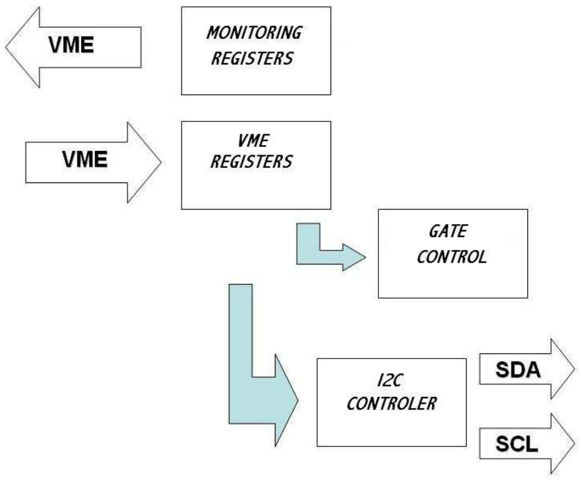

Figure 5: *LTPIM* control slave simplified schematic

The *I2C* bus communication developed by Philips(1) uses only two lines. There are the *Serial Clock (SDC)* and the *Serial Data (SDA)*. This bus however simple in structure provides a very reliable communication and since it only uses two lines is quite simple to layout.

The *I2C* communication is used to program a four independent channel *DAC* that will provide the analogue voltage used by the active equalizers.

It is also used to program the delay circuits on the *CTP* link output.

There are five registers that are needed to program the *I2C* communication, all accessed by *VME*.

The user can program the clock speed, and has total control of the bus communication at any time by *VME* access.

The *I2C* master core is an improved version of the Opencores version developed by Richard Herveille(2).

VI. TESTS

The tests done so far were in the prototype *FPGAs* and *VHDL* codes.

The *VME* interface was tested with success for all registers and for all *VME* bus configurations.

The *VHDL* control code was also tested in register write and read access and on the *I2C* communication with the slaves on the board. The hardware tests are undergoing on *CERN*, and despite a few problems regarding the active equalizers, they are proceeding well.

Hardware extensive tests as well as integration tests are still to be made.

VII. FUTURE WORK

Future work will address manly the high level software in order to program and control the *LTPIM*. There are some minor changes to be done on the *VHDL* codes in order to make then easier to understand. Schematic design will also suffer modifications before the *LTPIM* board goes into production.

VIII. LTPIM

All files regarding the *LTPIM* are available on the *CERN's EDMS* service.

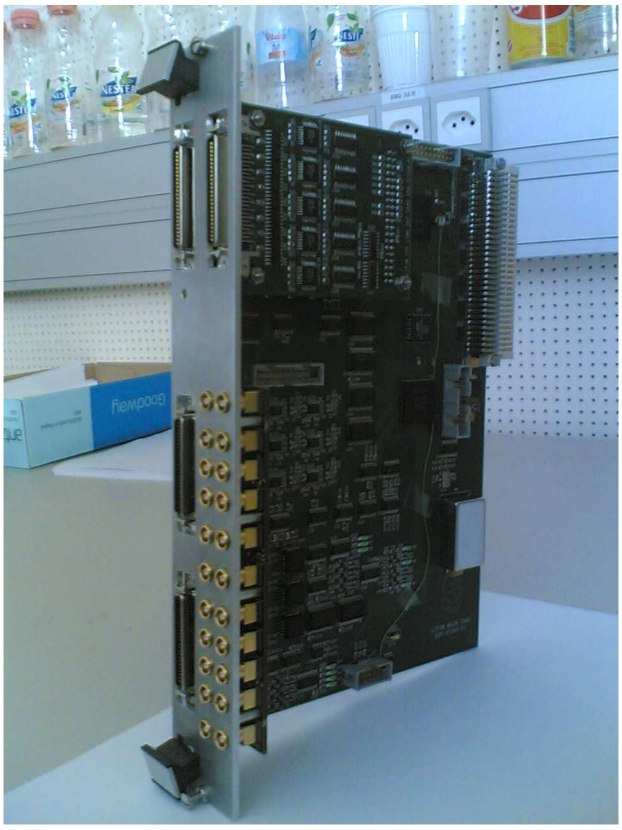

Figure 6: *LTPIM* Picture

IX. ACKNOLEDGES

We would like to thanks to Marcelo Perantoni and all of level one trigger team at *CERN* for all the excellent work and help during this project…

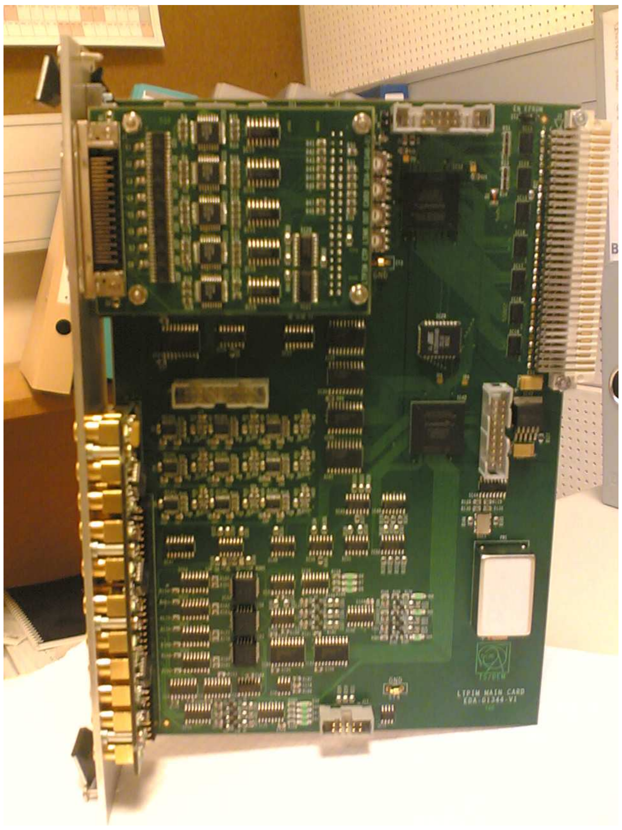

Figure 7: *LTPIM* Picture

X. . BIBLIOGRAPHY

(1) http://www.nxp.com/products/interface_control/i2c/

(2) www.opencores.org/Richard Herveille