

A Small Portable Test System for the TileCal Digitizer System

A. Hidvégi ^a, D. Eriksson ^a, C. Bohm ^a

^aStockholm University – Physics Dept., SE 10691 Stockholm, Sweden

attila@physto.se
daniel@physto.se
bohm@physto.se

Abstract

The hadronic Tile Calorimeter (TileCal) of the ATLAS detector at LHC has a digitization, pipeline and readout system composed of nearly 2000 boards [1][2], developed and maintained by Stockholm University. Prior to now a rather complex test system been used to verify the functionality of the boards. However this system was developed nearly 10 years ago and now difficult to maintain due to several already obsolete components. A new, simpler, more reliable, and portable test system was therefore initiated. Its components have been chosen to reduce problems with obsolescence, and to allow easy migration to new platforms over the lifetime of the digitizer system.

I. LEGACY SYSTEM

The original system for testing digitizer boards during development and manufacturing (Figure 1) is now over 10 year old. It contains several key components that would be difficult to replace in case of failure. The VME computer that controls the TTC system, for instance, has an obsolete architecture and operating system. The PMC optical receiver board that it hosts cannot be installed on many new systems. And if it can, there is still the problem of finding suitable drivers and subroutine packages. The aging hardware is causing increased instabilities. The documentation does not answer all questions. And perhaps most importantly, most of the persons that were involved in the development of the system over a long period are no longer available.

The legacy test system was intended for in-house testing and debugging of digitizer hardware in Stockholm. Ideally we want a more modern, practical test system that can even test the digitizer system *in situ*.



Figure 1: The legacy test system: complex and obsolete.

II. NEW SYSTEM

The new test system is based on commercially available components, for easy upgrade and service. It consists of a Xilinx development board (ML506) [3] (Figure 2), equipped with a Small Form-factor Pluggable (SFP) module for optical transmission and reception, and an optional laptop PC. The system communicates with the digitizers by transmitting TTC signals and receiving high-speed (G-Link) readout signals through the same SFP module, utilising a single gigabit transceiver on the Virtex-V FPGA (Figure 3). The FPGA can easily be reconfigured for future upgrades and improvements, such as hardware acceleration for different high-speed tests. It also contains an embedded CPU system running test software on Linux [4]. Hence we have replaced the previous crate system with a single board.

The board communicates with a host computer over Ethernet, allowing the debug software to be run remotely. The main debug software runs in a virtual environment for easy maintenance and compatibility.

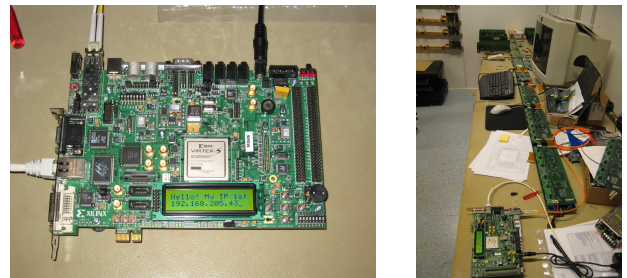


Figure 2: Xilinx ML506 development board (left) and full test system, including a complete set of drawer electronics (right)

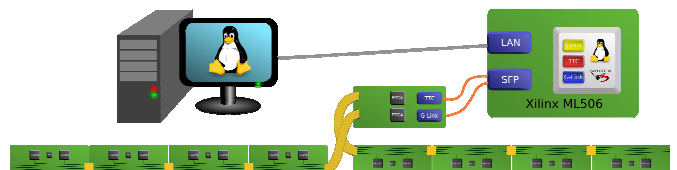


Figure 3: Diagram of the new test system.

III. DESIGN CHALLENGES AND SOLUTIONS

To create this system several design challenges had to be solved, including FPGA emulation of the TTC system and the Agilent G-link receiver.

A. Generation of the TTC signal

All configuration and command signals are transmitted to the digitizer system via the TTC system [5]. Therefore the TTC system needed to be emulated inside the FPGA. Currently the system uses a 100 MHz crystal oscillator to generate the 40 MHz clock on the receiver side, but the board could be equipped with an extra crystal oscillator to generate the 40.08 MHz clock too.

B. Receiving the G-Link signals

The digitizers transmit data to a link card, which in turn transmits them to the readout system over a high-speed Agilent G-Link [6] communication link. Therefore a G-Link receiver emulator was developed for FPGA as well, using one of the multi-gigabit transceiver GTPs included in the Virtex-V. Since the G-Link protocol is different from other common protocols on the market, some tricks with the GTP were needed: A Clock Data Recovery unit (CDR) is used to lock the receiver to the source frequency and the G-Link idle pattern is used as a comma character, to make the receiver align to the right bit pattern. The G-Link decoding is then implemented in normal logic cells. However, since the board has only one SFP module, both TTC and the G-Link needed to share the same GTP resource.

C. TTC and G-Link through the same SFP module

Transmitting TTC and receiving G-Link through the same SFP module is not supported by the module vendors, since the TTC is using 1300 nm wavelength and the G-Link is using 850 nm. Usually one SFP module supports only the same wavelength on both transmitting and receiving side. However, due to the good quality of optical transceivers on both sides the communication is very stable in both directions, even though they are used outside their specifications.

D. Embedded system running Linux

To control the TTC transmitter, the G-Link receiver and the network communication, an embedded CPU system running a variant of Linux was implemented to expedite porting of the legacy controlling software.

E. Porting the control software to the embedded Linux

Porting the control software to the embedded Linux was perhaps the most time consuming part of the work. In the new system the CPU has direct access to both the TTC transmitter and G-Link receiver. That means that transferring instructions and data between the debug software and the TTC transmitter and G-Link receiver should be quite simple. However, since the old hardware was much distributed, the controlling software was far more complex. Removing unnecessary parts

from the legacy software thus became the major task. The fact that many collaborators added redundant features to the old software in various places over a long period made porting more complex than necessary. Incomplete documentation also made the process more difficult.

IV. LONG TERM MAINTENANCE

The digitizer system will potentially need to function and be maintained for up to 15 years, so the test system must be reliable and robust. The ML506 development board could potentially break down at some point during this period. However, the board is commercially available and could easily be replaced with a new one without significant testing and debugging. If the board becomes unavailable after some years it could be replaced with a newer generation, since the necessary portion of the design is written in VHDL and could easily be implemented on a new FPGA. It is expected that the essential functionality (such as high speed links and fiber optics) will still be available. Generating a new embedded system and a Linux system will almost certainly be easier than for the current system.

The PC, of course, cannot be expected to be in use for 15 years. However, running the debug software in a virtual operating system makes it independent from the hardware and the operating system. (Figure 4) Virtualization technology will be maintained in the future by software developers and will almost certainly remain freely available. This eliminates the need for software maintenance from our side, by allowing us to retain the old operating system on any new hardware.

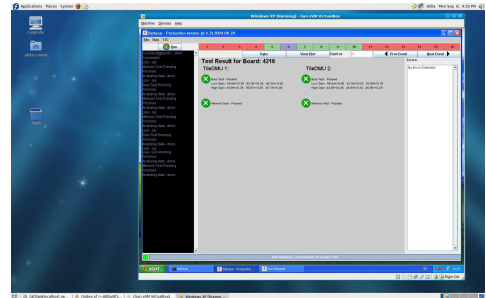


Figure 4: Debug software running in a virtual environment.

V. SUMMARY

The system is ready and more robust than the legacy system ever was. Important issues about future maintenance have been addressed and solutions found. Design challenges have been solved, although some developments took longer than expected.

Such a test system could be adapted for use in other systems, requiring only some amount of additional software development for the board. Even the hardware could be modified if communication links other than TTC and G-Link were desired.

VI. REFERENCES

- [1] <http://www.sysf.physto.se/atlas/digitizer>
- [2] <http://hep.uchicago.edu/atlas/tilecal/>
- [3] <http://www.xilinx.com>
- [4] <http://www.petalogix.org>
- [5] <http://ttc.web.cern.ch/TTC/intro.html>
- [6] Part #: HDMP-1032/1034 (can be found via Google)