An innovative telescope control system architecture for SST-GATE telescopes at the CTA Observatory

Gilles Fasola*[a], Shan Mignot[a], Philippe Laporte[a], Abdel Abchiche[b], Gilles Buchholtz[b] and Isabelle Jégouzo[a] for the CTA Consortium[c]

A prototype

SST-GATE, part of the GATE project, is a prototype for the Small Size Telescopes (SSTs) of the Cherenkov Telescope Array (CTA). Its purpose is to evaluate the performances of the previously untried Schwarzschild-Couder optical configuration. We have decided to take advantage of this opportunity to experiment several options for the design of the Telescope Control System (TCS).

These options are mainly related to some features of CompactRIOs (cRIOs) and LabVIEW: an embedded FPGA, the OPCUA protocol and the EtherCAT field-bus.

Communications

An overview of both the software and the communication layout of the TCS is shown in Figure 1:

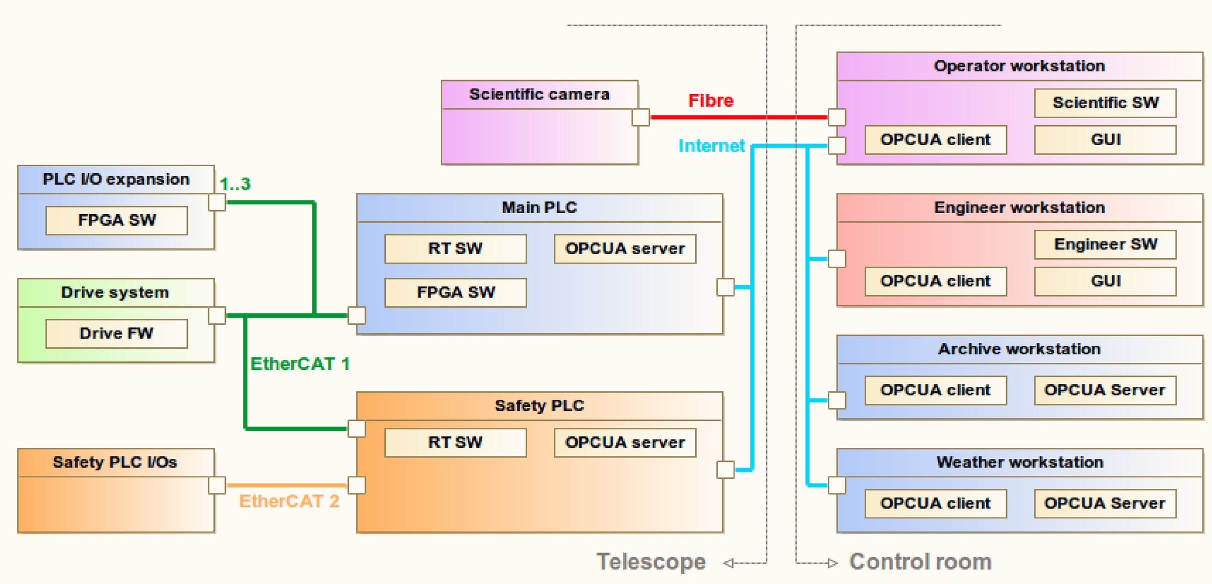

Figure 1: There are only two physical links between the telescope and the control room for communication purposes. A dedicated EtherCAT field-bus is used for safety. The logical connectivity is managed by OPC UA clients and servers. The network devices, such as switches, are not shown. HW stands for hardware, SW for software and FW for firmware.

EtherCAT for a real-time field-bus

The trend in industrial Ethernet technologies induces the use of only one network — both for safety-relevant and standard data — to save design effort and installation costs. In a first phase, there are two separate networks for the SST-GATE prototype (see Figure 1) to ease the development of each subsystem.

OPC UA as the software communication layer

The CPU load of a cRIO 9074 has been monitored versus several parameters. A sample of the results is presented in Figure 2:

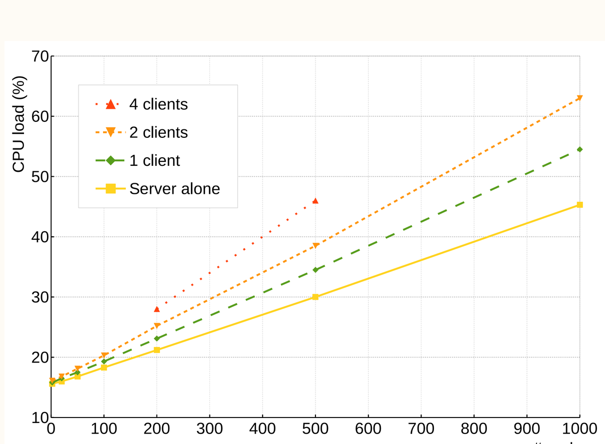

Figure 2: The CPU load of a cRIO 9074 while running an OPC UA server. The node values are published by the server and monitored by the clients every second.

Telescope pointing with an FPGA

Pointing computations

In CTA, each telescope is expected to point and track on its own based on the locally available computing resources. These reside in the cRIO and consist of real-time software running on the controller's processor and the programmable hardware: the FPGA.

To point towards a source, two types of transformations need to be computed: astrometric ones to account for the celestial motions and geometric ones for the imperfect geometry of the telescope. The required operators for these can be reduced to addition, multiplication, division and trigonometric functions: cosine, sine, tangent and their inverses.

CORDIC algorithms

To achieve optimised numerical computation with the embedded FPGA, LabVIEW provides a number of Intellectual Property (IP) cores, notably division, sine, cosine, tangent and the arc tangent functions. However, the arc sine and arc cosine functions are missing and have had to be developed in VHDL using LabVIEW's *Component Level IP* (CLIP) interface.

Suitable algorithms for these development exist based on the COordinate Rotation DIgital Computer (CORDIC) invented in 1959 by Jack E. Volder [1] for real-time airborne computation.

We have relied on the work of Lang et al. [2] — which is an improved extension of CORDIC for the computation of arc sine and arc cosine — to develop a VHDL component. It was simulated using ModelSIM (Figure 3) from Mentor Graphics for debugging, and synthesised to ensure all constructs were correctly inferred.

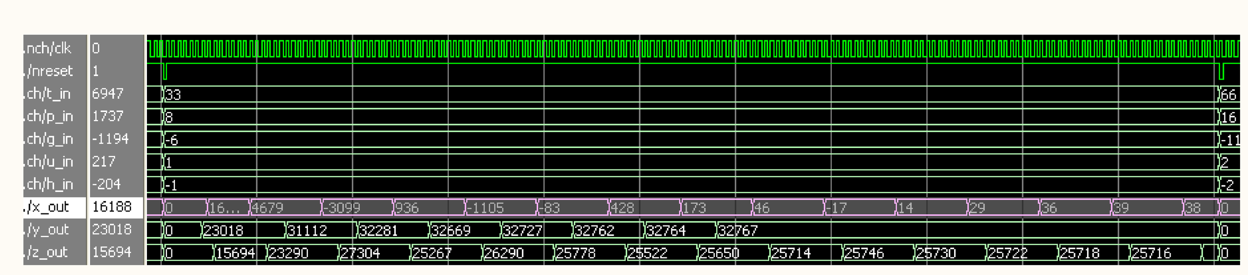

Figure 3: Snapshot of the first iterations of the approximate CORDIC algorithm as simulated by ModelSIM (with 16-bit words in this example). The different entries represent the ports of the component.

A VHDL component in a CLIP

From an architectural point of view, beside a testbench and a wrapper, the following architecture has been adopted:

- a top-level entity which provides the following services:
  - scheduling of operations within the IP
  - maintaining an iteration counter
  - determining whether the optimised computation should be used
- a module to compute the auxiliary variables
- the *XYZ module* which computes the output values

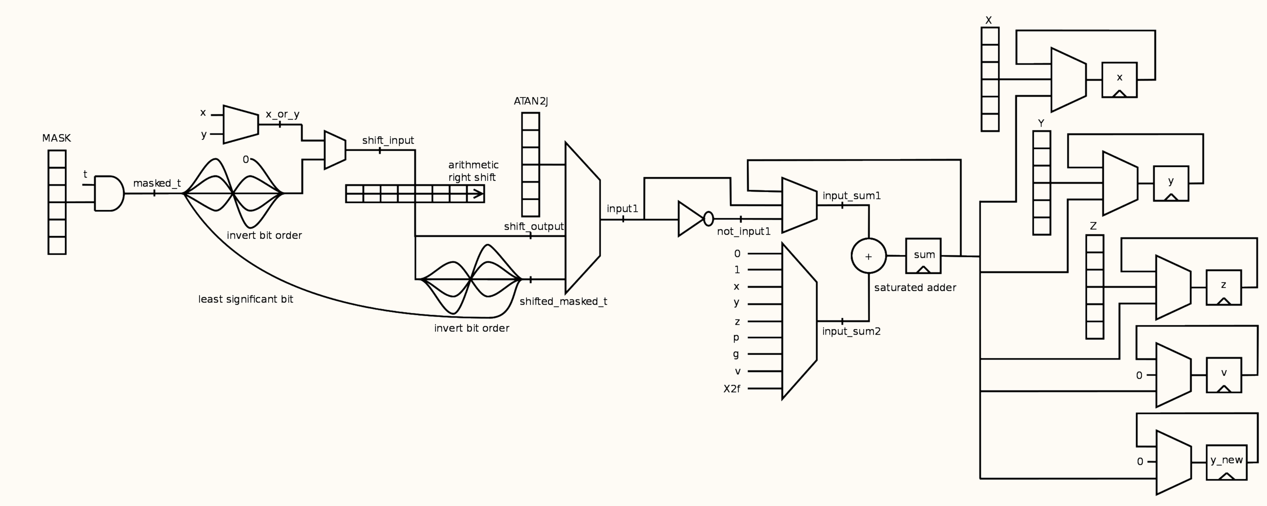

Figure 4: Register-transfer-level (RTL) representation of the XYZ module. This diagram does not show the control logic. Upper case names designate signals read from the read-only memories inside the FPGA (the vertical arrays).

An arc cos VI

A Virtual Instrument (VI) has been developed to hide the details of its wiring to internal RAM and of its timing from the user. The following table shows that our design, although still preliminary with its 16-bit words, is realistic as regards the target's resources both in terms of logic and timing.

Final device utilisation

Category	Used	Total	Percent
Slice registers	824	40960	2.0
Slice LUTs	1322	40960	3.2
Block RAMs	1	40	2.5

Final timing (post place and route)

Clocks	Req. (MHz)	Max. (MHz)
Auxiliary Clk	33.00	63.56
40 MHz Clk	40.00	68.18

The first results

The design of the TCS for SST-GATE is not completed yet. However, we can already report our first results. Our attempt to manage real-time tracking with an FPGA is about to be made conclusive. We have implemented the VHDL model of a CORDIC algorithm and have linked it to a real-time routine, all in a CompactRIO. The trigonometric functions being the main computation issue, elaborating the whole pointing algorithm is now just a matter of time.

Our choice of a CompactRIO as the core PLC of a telescope seems valid. We have checked the ability of this system to handle all of the required functions of the TCS individually: real-time telescope pointing, real-time access of the field devices and remote management via an industrial protocol.

Contact and Institutes

*) E-mail: gilles.fasola@obspm.fr
a) Galaxies, Étoiles, Physique et Instrumentation (GEPI) Observatoire de Paris – CNRS – Université Paris-Diderot, 5 place Jules Janssen, 92190 Meudon, France;
b) Division Technique de l'INSU (DT-INSU), CNRS, 1 place Aristide Briand, 92190 Meudon, France
c) www.cta-observatory.org

References

[1] Volder, J. E., "The CORDIC Trigonometric Computing Technique," *IRE Transactions on Electronic Computers* **EC-8**, 330–334 (1959).
[2] Lang, T. and Antelo, E., "CORDIC-based computation of arccos and $\sqrt{1-t^2}$," *Journal of VLSI signal processing* **25**, 19–38 (2000).

Acknowledgements

This work has been funded under the Convention 10022639 between the Région Île-de-France and the Observatoire de Paris. We gratefully acknowledge the Région Île-de-France, the CNRS (INSU and IN2P3), the CEA and the Observatoire de Paris for financial and technical support. We gratefully acknowledge support from the agencies and organizations listed in this page: https://portal.cta-observatory.org/public/Pages/FundingAgencies.aspx/. We gratefully acknowledge National Instruments France for technical support.