

# A NEW APPROACH TO THE ALFA TRIGGER SIMULATOR\*

BARTOSZ DZIEDZIC, KRZYSZTOF KORCYL

The Henryk Niewodniczański Institute of Nuclear Physics  
Polish Academy of Sciences  
Radzikowskiego 152, 31-342 Kraków, Poland

*(Received April 13, 2016)*

The idea and principle of operation of a device which is used to test the trigger system of the ALFA detectors of the ATLAS experiment at the LHC are discussed. A new approach to the application control is presented. The application runs under control of the ArchLinuxARM operating system. Also, the drawbacks of the new solution are discussed.

DOI:10.5506/APhysPolB.47.1655

## 1. Introduction

During the Long Shutdown 1, the LHC underwent several modifications. In addition, the detectors and the infrastructure were improved. The ALFA detector system [1] of the ATLAS experiment [2] received a new trigger interface called the ALFA\_CTPIN, which is used to transmit the ALFA trigger signals to the ATLAS Central Trigger Processor (CTP). To ensure that the new instrumentation works correctly, comprehensive tests have to be carried out. This was the motivation beyond the development of the hardware ALFA Trigger Simulator (S3T) [3].

The solution presented below is also based on the Zynq Evaluation and Design Board — the ZedBoard. The ZedBoard is equipped with modern System-on-Chip (SoC) integrated circuit Zynq Z7020 [4]. The Zynq chip contains a micro-controller with the dual-core ARM Cortex A9 processor and the programmable logic part (FPGA), based on the Artix 7 architecture [5].

In 2015, both the FPGA firmware (logic) and the S3T software were changed. The firmware changes reflect a new idea of the ALFA\_CTPIN tests are described in Sec. 2. Section 3 is devoted to the description of

---

\* Presented by B. Dziejdzic at the Cracow Epiphany Conference on the Physics in LHC Run 2, Kraków, Poland, January 7–9, 2016.

the FPGA firmware changes. The software changes mostly result from the requirement of the remote LAN access and migration from an independent standalone application to the Linux application. The new approach provides also a possibility to upgrade both the software and firmware without physical access to the board. Some changes are the result of the firmware upgrade. A full discussion of the software changes is presented in Sec. 4.

## 2. ALFA\_CTPIN examination

First ALFA\_CTPIN tests relied on the direct, hardware connection between the S3T and the CTP interface (Fig. 1). This solution offered a possibility to check the CTP response to the trigger signals which had come from the Main- and Overlap-Detectors (MD and OD) of the ALFA. Presently, the logic implemented in ALFA\_CTPIN allows performing such tests using the preprogrammed input memory. However, the S3T can still be used for the hardware simulation of the MD or OD pulses.

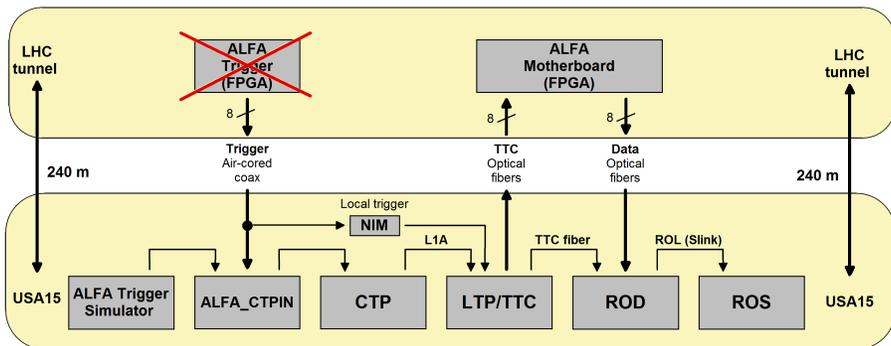


Fig. 1. Scheme of direct connection between S3T and ALFA\_CTPIN [6].

In the discussed approach, the main idea of the test has changed. The cables installed in the LHC tunnel to originally transmit OD signals from the stations to the control room can now be used to connect the S3T to the test LEDs located inside the Roman Pots (Fig. 2). This change required that the firmware installed in the PMFs should be changed to be able to receive NIM signals from the control room and convert them to pulses illuminating LEDs in the detector stations. With such wiring, the signals from the S3T can force the LEDs to blink. The light emitted by the LEDs excites the trigger photomultipliers and, in this way, the detector trigger signals are generated. In the new series of tests, only the Main Detectors will be tested since the trigger logic gives a higher priority to the MD signals over the OD. Hence, in such a way, mostly the MD triggers are generated, OD will only fire on inefficient flash on MD.

This kind of test involves the whole signal chain — from the detectors through the multi-anode photomultipliers, the front-end electronics, the air-core cables up to the interface mentioned before.

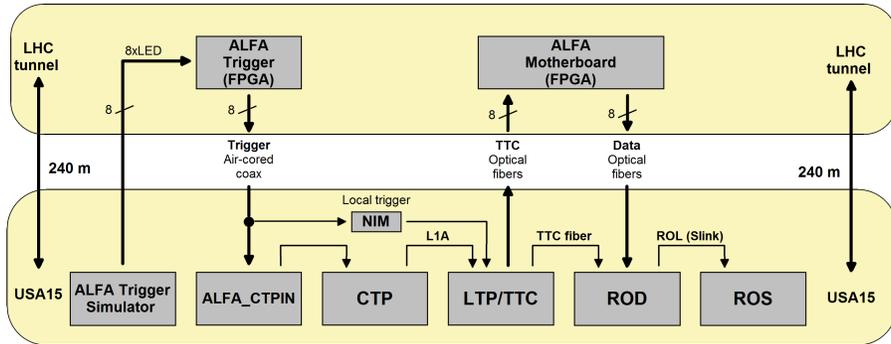


Fig. 2. Connection between S3T and LEDs inside of Roman Pots [6].

### 3. Architecture of S3T version 4.x

A block diagram of the S3T version 4.x architecture is presented in Fig. 3. Since in the presented approach only the Main Detector signal chain will be tested, there is no need to multiply the S3T operational frequency to 80 MHz which was used in the previous version [3] to simulate the 12.5 ns long ODs signals. The Xilinx module of Clocking Wizard (a block converting and managing the clock signals) is still part of the design as it is used to switch between different sources of the clock: local or from the LHC.

Similarly to the old approach, the user will use a console to manage the test parameters, configure the table of patterns which will be generated by the S3T and decide when the test should be started. The data and parameters are transferred from the ARM program to the FPGA core through the AXI Lite bus. The data are sent to the BlockRAM (the dual port RAM memory located in the FPGA) using Port A. The user can also set the clock signal source to an internal clock for the S3T tests and debugging, or to the external (the LHC clock) one for the ALFA\_CTPIN examination. In both cases, the frequency is 40 MHz. One should note that the LHC clock frequency is 40.07897 MHz [8]. The ORBIT signal (the 11.2455 kHz signal which is the beam revolution frequency at the LHC [8]) is used to reset the counter, and to synchronise the bunch number and the BlockRAM memory cell. Additional delay and dummy orbit signals (the orbit signal cycles without generating the output signal) can be configured.

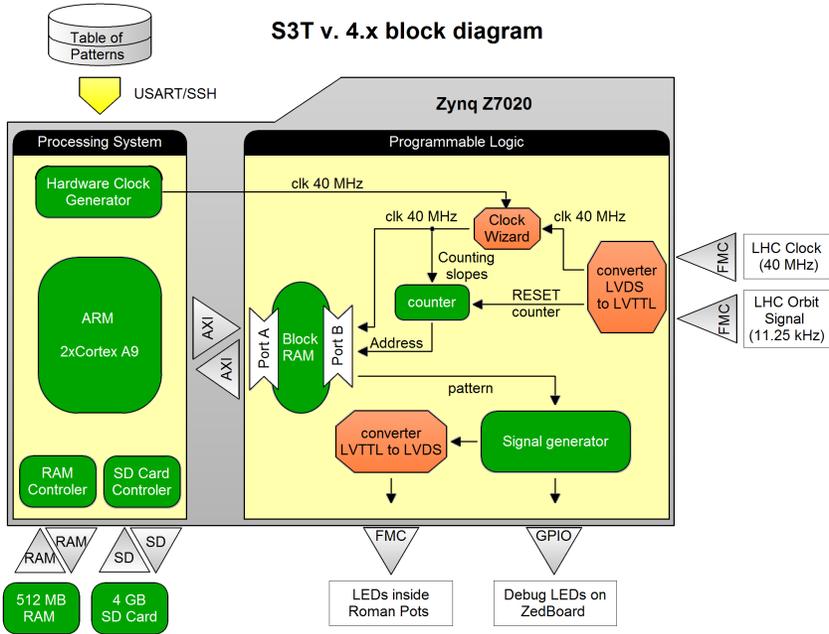


Fig. 3. Block diagram of S3T [3].

The basic working mode is a synchronous operation with the LHC clock. The counter counts the LHC clock pulses and addresses the BlockRAM memory. The signal generator reads the pattern from the memory via Port B, using the address pointed by the counter, and accordingly generates the output signal with optional, additional parameters, such as the dummy orbit or delay.

#### 4. Software

A previous version of the software allowed the user–device communication only via the USB port (a built-in USB/USART converter). This was a basic solution, but for the remote device control, it required an additional computer connected to the ZedBoard via a dedicated Digilent USB connection. This solution was both impractical and inconvenient. Moreover, every update required physical access to the board in order to reboot the system or to replace the memory card.

In the new edition of the software, the standalone application (an independent application operating on Zynq) has been replaced by an application which works under control of the ArchLinuxARM operating system. One should note that this system is the installation ready for the ZedBoard devices.

The operating system allows many applications including the ssh or web servers to be used. The former deliver the communication platform with which the user can control the board remotely. This is possible because the ZedBoard is equipped with the ethernet socket and the Zynq implements the hardware Gigabit Ethernet controller [4, 5]. ArchLinuxARM gives full access, via the ssh protocol, to the system control at the system administrator level. This makes it possible to run the application with the requirement of the root privileges, in particular, those necessary to configure or reconfigure the FPGA or to edit the file system. In addition, the user can employ various versions of software and firmware in accordance with the currently executed test. The user can change the firmware at any time without physical access to the board and without restarting it.

A block diagram of the data exchange (commands, settings, table of patterns) between the application and FPGA is shown in Fig. 4. By using the mknod command, the user can create a special file called `/dev/xdevcfg`, which represents the character device. Next, the user sends the prepared FPGA configuration file (bit format) to `/dev/xdevcfg`. The Xilinx driver recognizes this special file by its name and eventually configures the FPGA logic. Then, the user can run the control application.

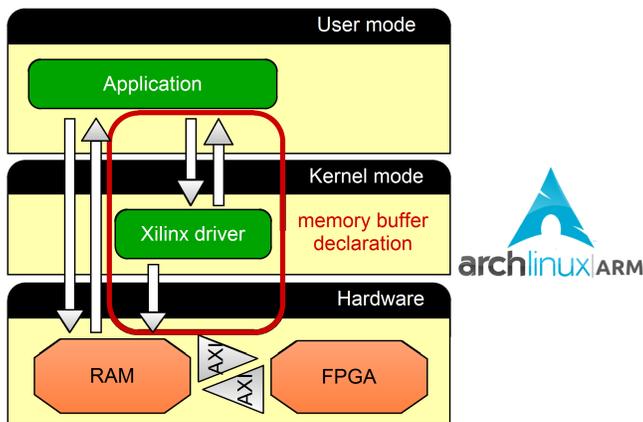


Fig. 4. Memory mapping under Linux.

The application accesses the FPGA part of the Zynq via the AXI Lite bus using the memory mapping. The connection using the memory mapping is based on a memory buffer located in RAM which can be accessed by both the FPGA and the S3T software. To make it possible, the designer has to use the same set of registers during the firmware design and later during the application creation. In this scheme, the FPGA works as a slave and only the application (which is a master) can establish the connection. The

FPGA sends the data to the application only on demand. Because of the lack of a direct communication between the software and the firmware (the communication uses the memory buffer), there are additional delays in the application performance. Owing to the fact that the FPGA firmware works with 40 MHz frequency and the ARM operates at 667 MHz clock, they have to be synchronised. For this purpose, a dual link BlockRAM is used with one port (Port A) for the memory communication via the AXI bus, and the second one (Port B) serving the Signal Generator in FPGA.

## 5. Summary

The application of the ArchLinuxARM operating system to the ZedBoard increased the flexibility of the hardware in terms of functionality and communication capabilities. Through the use of Linux, the remote access and communication with the S3T LAN network became a reality. Communication via USB/USART is still possible. Linux extends the functionality of software update and logic reconfiguration, which has a great influence on the S3T development.

This work was supported in part by the Polish National Science Centre grant UMO-2015/18/M/ST2/00098. The ZedBoard purchase was financed by the Polish National Science Centre Grant UMO-2012/05/B/ST2/02480.

## REFERENCES

- [1] ATLAS Collaboration, ATLAS Forward Detectors for Measurement of Elastic Scattering and Luminosity, CERN-LHCC-2008-004.
- [2] ATLAS Collaboration, *JINST* **3**, S08003 (2008).
- [3] B. Dziedzic, K. Korcyl, *Acta Phys. Pol. B* **46**, 1271 (2015).
- [4] AVNET, ZedBoard — Hardware User's Guide.
- [5] Xilinx, Zynq 7000 All Programmable SoC — Technical Design Report.
- [6] K. Korcyl, ALFA Detector: Timing and Trigger, Proc. of SPIE, 89032K-1.
- [7] B. Allongue *et al.*, *JINST* **7**, C02034 (2012).
- [8] B.G. Taylor, Timing Distribution at the LHC, 8<sup>th</sup> Workshop on Electronics for LHC Experiments, Colmar, September 9–13, 2002.