Contents lists available at ScienceDirect

Nuclear Inst. and Methods in Physics Research, A

journal homepage: www.elsevier.com/locate/nima

A Trigger Interface Board for the Large and Medium Sized telescopes of the Cherenkov Telescope Array

Luis A. Tejedor [a,b,*], Juan A. Barrio [a,b], Pablo Peñil [a,b], Alejandro Pérez [a,b,c], Diego Herranz [d], Jorge Martín [b]

[a] *IPARCOS Institute, Universidad Complutense de Madrid, E-28040 Madrid, Spain*
[b] *EMFTEL Department, Universidad Complutense de Madrid, E-28040 Madrid, Spain*
[c] *IFAE Institute, E-08193 Bellaterra, Barcelona, Spain*
[d] *FAMN Department, Universidad Complutense de Madrid, E-28040 Madrid, Spain*

ARTICLE INFO

Keywords:
CTA
Camera
Cherenkov
Gamma-ray
Astrophysics
Trigger
Stereo
Firmware
OPC-UA
Busy
State machine

ABSTRACT

This paper presents the Trigger Interface Board (TIB), a system in charge of managing the trigger and clock signals in the cameras for the Large- and Medium-Sized telescopes in the Cherenkov Telescope Array (CTA). The TIB includes interfaces with the neighboring telescopes and with other subsystems in the camera. It receives trigger pulses from different origins and decides when those trigger signals should cause the camera readout. The decision is based on slow control parameters, the TIB internal state, the busy state of the camera, and the coincidence in time with other telescopes in a hardware stereo trigger scheme. Depending on the trigger origin, the TIB also generates an event data fragment, that is delivered to the camera Event Builder as part of the full raw event data. Additionally, it provides camera data-taking information to CTA central instances for monitoring purposes. The design of the TIB is based on two main components, namely an FPGA and a microcomputer. This structure provides fast digital processing capabilities, high-level software features for slow control and TCP/IP communication, and, especially, great flexibility to develop functionalities by working on the firmware and software levels separately.

1. Introduction

The Cherenkov Telescope Array (CTA) Consortium [1] is building two arrays of Cherenkov telescopes, one on each hemisphere. Its target is to create an all-sky observatory to characterize the cosmic γ-ray photons that reach the atmosphere by measuring the Cherenkov light emitted by the γ-ray atmosphere interaction [2], improving the sensitivity and angular resolution of the current Cherenkov telescopes by an order of magnitude [3]. In order to reach this goal over a wide energy range, each array will contain telescopes with different characteristics. The Large-Sized Telescopes (LSTs), 4 of them arranged in a 100 m square, are optimized to detect Cherenkov showers caused by γ-rays between 10 and 300 GeV. The Medium-Sized Telescopes (MSTs), up to 25 covering one square km, are designed to improve the sensitivity in the core CTA energy domain, between 150 GeV and 10 TeV. Finally, the Small-Sized Telescopes (SSTs), up to 70 of them, distributed in a 5 square km grid, are designed to provide a maximum sensitivity between 10 and 300 TeV. The subsystem described in this paper, known as Trigger Interface Board (hereinafter TIB), and previously presented in [4,5], will be installed in all the cameras of the LSTs [6] and NectarCAM MSTs [7], where it will play a central role.

These cameras follow the classical acquisition scheme based on switched capacitor arrays (SCAs). The analog signals detected by the photosensors are sampled continuously at a 1 GHz rate, storing the analog voltage in a circular array of capacitors implemented in the DRS4 chip [8] (LSTCAM) or the NeCTAr Chip [9] (NectarCAM). The voltage samples stored in the capacitors are not digitized continuously, but only when the camera trigger system detects that an interesting event has taken place. In this case, a trigger signal is distributed to the acquisition channels corresponding to each pixel, the analog sampling is stopped, and several tens of samples around the trigger time are digitized. This scheme shows various advantages: first, the requirement of a fast digitizer per pixel is avoided, saving costs and power consumption. Second, one can avoid to digitize, process, and store a large amount of pixel information containing only noise. Considering that the LSTCAM and NectarCAM have 265 Front-End Boards (FEBs) of 7 pixels each, sampling continuously at 1 GHz and digitizing with 12 bits, it would mean a data stream of 2.78 TB/s, which would be very difficult to handle. However, this scheme has an inherent limitation: the camera is not able to sample and record the events while the digitization process is taking place. This effect is known as read-out dead time.

[*] Correspondence to: EMFTEL Dept., Physics Faculty Pza. Ciencias, 1, E28040, Madrid, Spain.
E-mail address: luistejedor@fis.ucm.es (L.A. Tejedor).

https://doi.org/10.1016/j.nima.2021.166058
Received 1 September 2021; Received in revised form 7 November 2021; Accepted 15 November 2021
Available online 17 December 2021

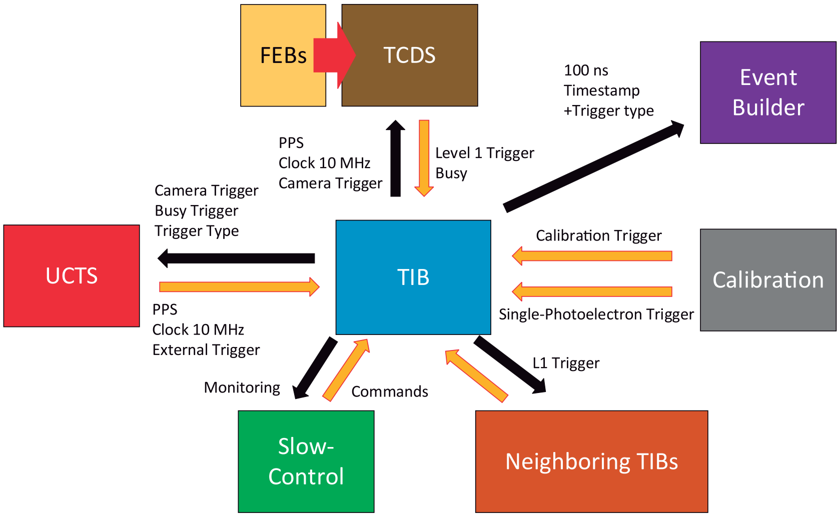


Fig. 1. Camera subsystems interacting with the TIB.

Thereby, the trigger system must be able to distinguish between interesting air-shower events in the camera and noisy events from the Night Sky Background light (NSB). The main difference between them is that the photons coming from Cherenkov showers reach the camera in correlated time and compact angular space, while the NSB photons arrive randomly. In the case of low-energy γ-rays, the light induced by the cosmic γ-ray is very faint, and the distinction between Cherenkov-shower images and NSB ones is difficult, requiring complex trigger systems [5,10,11]. In particular, the trigger system of the CTA LSTCAM and NectarCAM MSTs is structured in different levels:

- Level 0 corresponds either to discriminated individual pixel signals (NectarCAM) or to the analog sum of the 7 pixels connected to each FEB (LSTCAM).
- Level 1 performs different combinations of the L0 signals from neighboring clusters. If any of the evaluated combinations of clusters is receiving an excess of light intensity above a programmable threshold, it can be assumed that the camera has detected an event. Thus, a digital *L1* signal is generated, which, in turn, starts the reading-out of all the camera pixels.
- A higher, more sophisticated, camera trigger level, usually called Level 2, is implemented differently for LSTCAM and NectarCAM. In LSTCAM, the *L2*, also known as Hardware Stereo Trigger, looks for coincident detections by two or more telescopes in a certain time window (typically 50 ns). As the NSB photons arrive randomly, it is very unlikely to record NSB-induced noise events in two telescopes at correlated times. The LSTs operating in stereo mode only digitize the events detected by at least 2 coincident telescopes, rejecting most of the NSB events and thus improving the sensitivity to the weak camera images caused by low energy γ-rays. In NectarCAM, the *L2* is implemented by combining in an FPGA the *L1* signals from the 265 clusters, so that dedicated camera patterns maximizing γ-ray-like events with respect to NSB noise events can be searched for. For the sake of simplicity, hereafter both the *L1* and *L2* signals reaching the TIB will be labeled as *L1*.
- A final Level 3, operating at an array level and known in CTA as software array trigger (SWAT), looks for coincident timestamps corresponding to events digitized and stored in memory by all the telescopes in the array. Only events fulfilling an *L3* condition are finally saved on disk. The level 3 does not improve the telescope dead-time, avoiding digitizing noise events, but it reduces the data throughput. [12].

It is worth stressing that the time available for the trigger decision at levels 0, 1, and 2 is limited by the length of the analog buffer of the SCAs. If the trigger decision takes longer than the time that the analog signals are stored in the ring memories, the samples of the corresponding signals would be overwritten. Therefore, the longer latency introduced by a trigger scheme, the longer buffer is required. This is the main difference between the NectarCAM and the LSTCAM: the NectarCAM is designed to be triggered with a fast *L2*, with a latency smaller than 400 ns, so 1 μs-long buffer is enough. On the other hand, the LST *L2* is formed by a Hardware Stereo Trigger, requiring to exchange *L1* signals among neighboring LSTs, so the SCA of each pixel is equipped with 4 μs-long buffers.

Implementing the LST Hardware Stereo Trigger was the original purpose of the TIB. However, during the development of the LSTCAM and NectarCAM, the necessity of a central unit managing all camera trigger signals became evident, so the TIB powerful features, described below, allowed it to implement this additional functionality, which is essentially common for both LSTCAM and NectarCAM.

2. The TIB in the camera

The TIB interacts with many subsystems in the camera, both for LSTCAM and NectarCAM, as is shown in Fig. 1. As already commented, the TIB receives the *L1* signal from the Trigger and Clock Distribution System (labeled as TCDS in Fig. 1) and sends the *camera* trigger signal back to the TCDS when the camera needs to be read-out. Apart from the *L1* trigger pulses, the TIB takes into account many other configuration parameters, signals, and conditions to generate the *camera* trigger. For instance, if the LSTs are operating in hardware stereo mode, the TIB will take into account the *L1* signals from the neighboring telescopes. Additionally, it might be required to read the camera randomly when no *L1* trigger is generated (the so-called *Pedestal* triggers, which are used for image reduction purposes), Finally, for calibration purposes, the camera is illuminated either by bright light sources (so-called *Flat-Field Calibration*) or by dim ones (known as *Single Photo-Electron Calibration*) [13,14]. Therefore, the TIB must handle the corresponding *Calibration* and *Single Phe* triggers. All these different *camera* trigger sources are managed by the TIB.

On the other hand, whatever the trigger origin is, the TIB cannot send triggers to the FEBs while they are digitizing an event. The TIB is informed about this situation by means of the busy signal generated at the FEBs and centralized by the TCDS. The TIB is also not delivering *camera* triggers to the TCDS if the state machine is not in

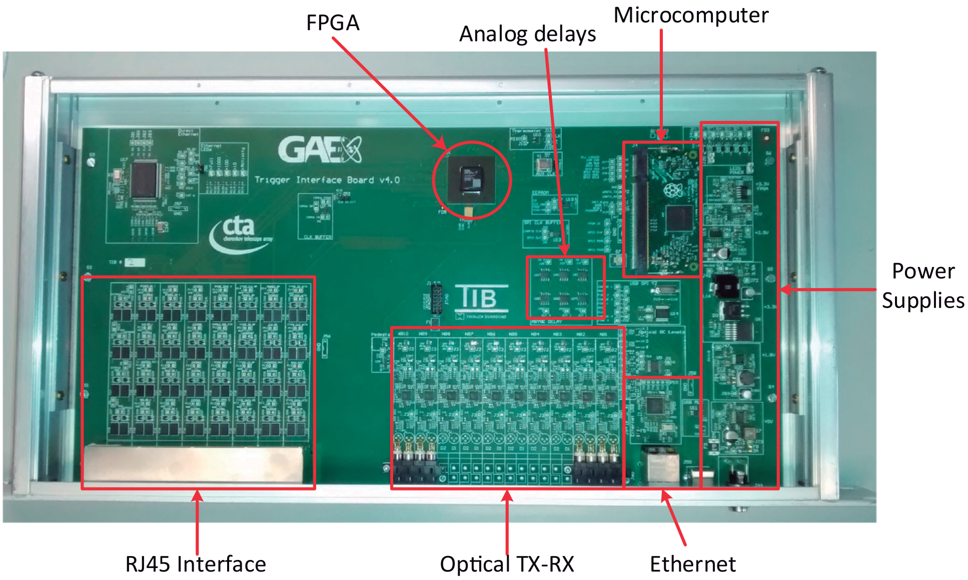


Fig. 2. Photograph of the TIB, with its main modules.

the correct state (e.g. the clusters cannot accept *camera* triggers while they are being reset or configured). The state of the TIB is configured by a global camera controller software (hereinafter CC) running on a dedicated computer and represented in Fig. 1 by the Slow-Control box. Additionally, the slow control interface of the TIB allows the CC to configure all TIB parameters, like the accepted trigger origins or the hardware stereo coincidence window length, and provides monitoring of trigger rates, TIB status, number of triggers during a run, etc.

Besides these functions, as it is shown in Fig. 1, the TIB also participates in the clock distribution and timestamping of events. The clock signal distribution in the CTA array is based on the White Rabbit technology [15]. Every telescope has a White Rabbit node, with the generic name of UCTS (Universal Clock and Time Stamping board), which provides a PPS (pulse per second) and a 10 MHz clock, both with ns accuracy. The clock and the PPS signals are delivered from the UCTS to the TIB continuously via an RJ45 interface, and the TIB controls when they should be delivered to the TCDS depending on the camera state. In addition, the UCTS timestamps the *camera* trigger signals delivered by the TIB, also with nanosecond accuracy [16,17]. Besides providing the *camera* triggers, the TIB assigns a bitmask encoding the actual trigger source (mono, stereo, calibration, pedestal, etc.), hereinafter *trigger type*, and the telescopes that participated in a hardware stereo trigger, hereinafter *stereo pattern*, associated with each *camera* trigger pulse. This information is delivered by the TIB to the UCTS, so the latter adds it to the timestamp of each event. Then, it delivers the full packet to the SWAT, so that it can perform the final Array Trigger *L3*.

Finally, in order to assure redundancy in the timestamp delivery, the TIB also generates its own 100 ns accurate timestamp which, together with the *trigger type* and the *stereo pattern* is sent directly to the Event Builder (EVB) through an Ethernet link. The EVB refers to the system in charge of receiving the data from every FEB, some configuration parameters, the *UCTS/TIB timestamps* and *trigger types*, and organizing all this information in a camera event structure [18].

3. Hardware structure

A photograph of the TIB, housed inside a 22 cm depth 1U rack box, is shown in Fig. 2. The main components are a microcomputer and a fast FPGA Xilinx Artix 7 with 676 pads. Due to the complex pinout of the FPGA and the necessity to handle weak analog signals, digital pulses, high-speed clocks, and different power rails on the same board, the PCB was designed with 16 layers, including 8 signal layers and 8 power planes. In this section, the main hardware components of the board are described.

3.1. Microcomputer

A microcomputer, with a Linux OS running the slow control server, transmits the corresponding commands to the FPGA or other simple devices like the thermometer or the asynchronous delay lines and reads the monitored parameters. The communication with the FPGA and the other devices is done via SPI protocol. A Raspberry Pi Compute Module v3 was chosen for this project due to the vast amount of available software, the large community using the Raspberry Pi platform, and the high reliability of this industrial version of the Raspberry Pi [19]. This microcomputer fits into a standard DDR2 SODIMM connector and includes neither power supplies nor an Ethernet interface. Instead, it uses the facilities of the TIB board, as shown in Fig. 2. In particular, the Ethernet interface is used for slow control communication and to send the *TIB timestamp* + *trigger type* + *stereo pattern* information to the EVB.

3.2. FPGA

An FPGA Xilinx Artix 7 [20] was chosen to handle the trigger signals and implement the logic which will be explained in the following sections. This FPGA was selected for its high speed, relative simplicity, and low power consumption. The 10 MHz clock arriving from the UCTS feeds the FPGA, which generates several internal frequencies as high as 400 MHz (i.e. 2.5 ns period), needed to handle the pulses with the required accuracy.

3.3. RJ45 and Ethernet

Fig. 2 shows a large space devoted to RJ45 interfaces and Ethernet. On the one hand, the TIB uses a matrix connector with 16 RJ45 positions for communications with other modules inside the camera. For instance, two RJ45 connectors are reserved for communication with the UCTS, while two others are occupied by cables coming from the TCDS. The signals arriving at these connectors do not follow the Ethernet standard, but they are just LVDS signals using the differential

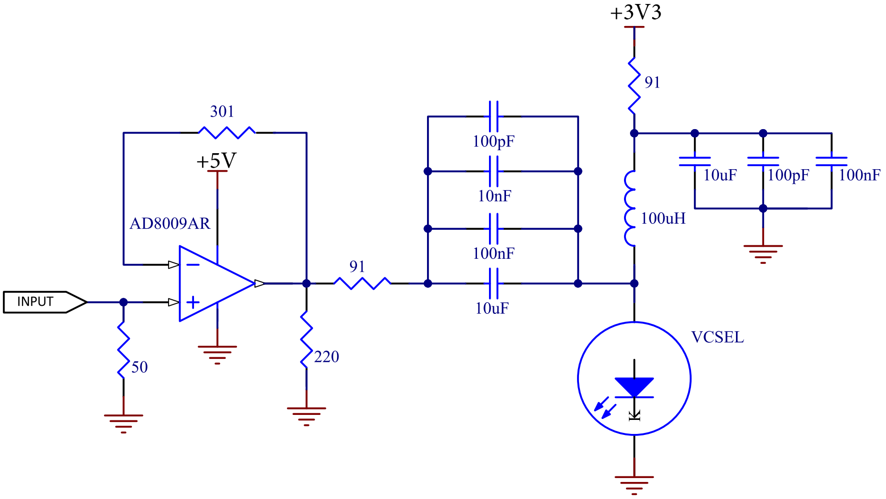


Fig. 3. VCSEL driver circuit.

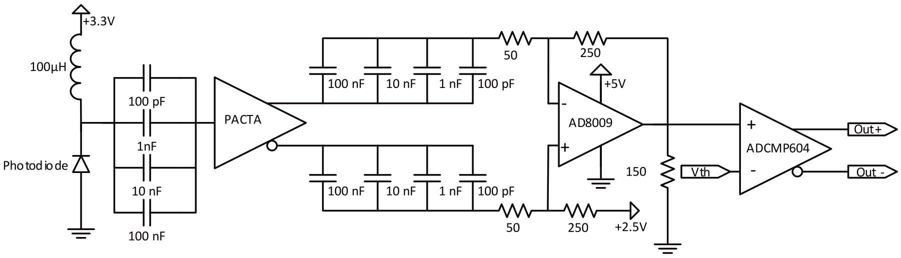


Fig. 4. Optical receiver.

pairs in a Cat. 6 Ethernet cable. These signals are handled directly by the FPGA and, as the FPGA is a critical part of the TIB, each input is protected with Schottky diodes and quenching resistors, using the large area marked in the photograph. On the other hand, the area labeled as Ethernet corresponds to a real Ethernet connection which is used by the communications between the microcomputer, the CC, and the EVB, which are, after all, computers.

3.4. Optical channels

When a TIB in a given LST receives a local *L1* trigger pulse, it is immediately replicated and sent to the neighbor LSTs using optical fibers. In the same way, the *L1* trigger pulses corresponding to each neighbor LST are received by the local one. High bandwidth is required to keep the accurate timing information encoded in the leading edge of these *L1* trigger pulses. To achieve it, a VCSEL and a PiN photodiode with 1 GHz bandwidth were selected for the Tx and Rx circuits respectively. Both operate at 850 nm and are assembled in housings following the optical LC standard [21]. The VCSEL model is the Truelight TMC-5A40-008 [22] and the photodiode model is the Hamamatsu S9573-01 [23]. As the maximum expected distance between TIBs is shorter than 500 m, multimode optical fibers 50/125 OM3 were used [24]. The TIB also uses the optical receiver circuit to collect the trigger signals delivered by the LST and NectarCAM Flat Field Calibration light sources. These light sources are placed at the center of the LST and MST reflectors and the corresponding trigger pulses need to travel typically 100 m and 50 m respectively. Therefore, the optical transmission of these trigger pulses is more adequate than the use of electrical signals.

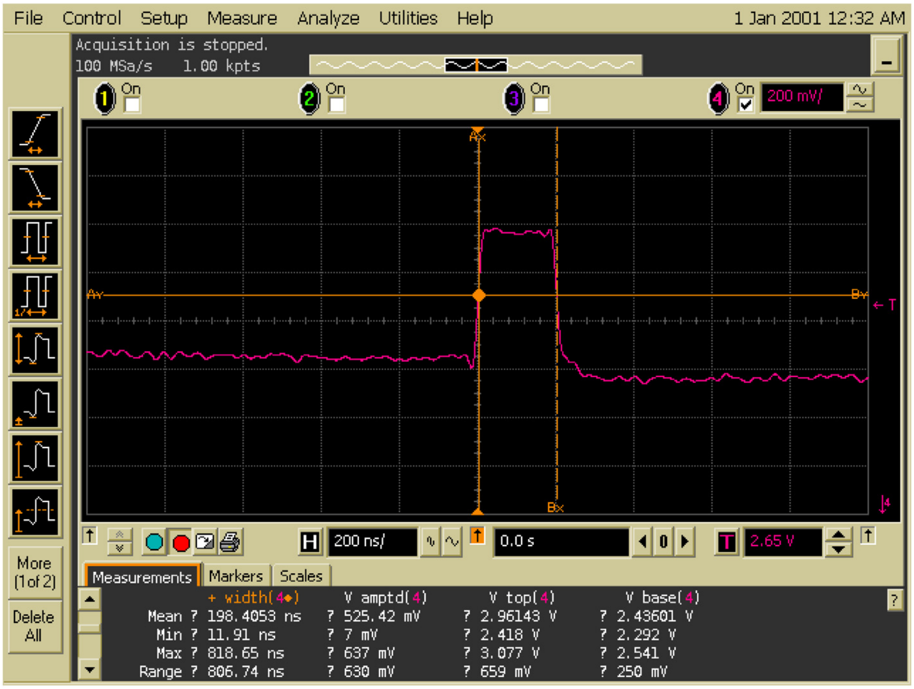
Apart from the optical stage, the electronics associated with the emitter and receiver must be able to handle pulses with typical widths of 200 ns and rise times of 5 ns. The circuit driving the VCSEL is shown in Fig. 3. The input signal comes from the FPGA in CMOS logic and the driver uses a fast operational amplifier AD8009 [25], with 1 GHz bandwidth. The gain of the circuit can be adjusted by replacing the 91 Ω resistor.

Regarding the receiver circuit, the current signal drained by the photodiode is amplified in a special transimpedance amplifier developed for CTA named PACTA [26]. The differential output voltage from the PACTA is amplified and translated to single-ended with the AD8009 configured as subtractor–amplifier, as is shown in Fig. 4. Finally, the single-ended signal is compared with a programmable threshold in a comparator ADCMP604 [27], which provides the LVDS signal that is delivered to the FPGA.

The tests in the laboratory with a 500 m OM3 optical fiber between the emitter and the receiver show a clear pulse of 0.5 V amplitude at the input of the comparator for a 3.3 V signal at the emitter input, as it is shown in Fig. 5. As there is only 1 LST built at the time of writing this paper, there are no field measurements available yet, although the lab set-up is expected to represent the final conditions.

3.5. Asynchronous delay lines

The LSTCAM readout employs a chain of DRS4 chips to implement the SCA rings, while the NectarCAM one is based on the NeCTAr chip [9]. The digitization of the samples stored in the SCAs begins when a *camera* trigger is delivered by the TIB. In order to reduce the dead time associated with the digitization of all the samples in the SCAs and the size of the events, the *camera* trigger is synchronous with a delayed copy of the incoming *L1* signal. This delay must be constant and with very low jitter, so the FEB can consider the leading edge of the *camera*



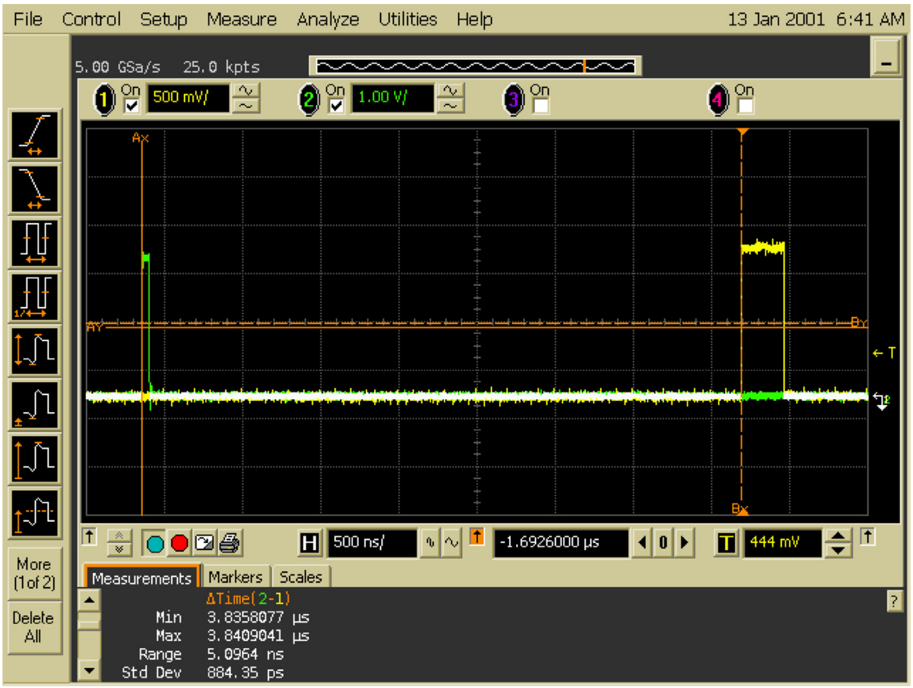
```
Acquisition is stopped.
100 MSa/s    1.00 kpts
```

4 ☑ On 200 mV/

200 ns/ 0.0 s 2.65 V

```
        + width(4•)       V amptd(4)     V top(4)       V base(4)
 Mean ? 198.4053 ns     ? 525.42 mV    ? 2.96143 V    ? 2.43601 V
  Min ? 11.91 ns        ? 7 mV         ? 2.418 V      ? 2.292 V
  Max ? 818.65 ns       ? 637 mV       ? 3.077 V      ? 2.541 V
Range ? 806.74 ns       ? 630 mV       ? 659 mV       ? 250 mV
```

Fig. 5. Received analog pulse at the input of the comparator.



```
5.00 GSa/s    25.0 kpts
```

1 ☑ On 500 mV/ 2 ☑ On 1.00 V/

500 ns/ -1.6926000 µs 444 mV

```
          ΔTime(2-1)
     Min  3.8358077 µs
     Max  3.8409041 µs
   Range  5.0964 ns
 Std Dev  884.35 ps
```

Fig. 6. Delay and jitter measured for the default 3825 ns delay value.

trigger as a time reference, useful to digitize the samples corresponding to the γ-ray shower image. In this way, digitizing around 40 samples in LSTCAM and 60 in NectarCAM is enough to record the complete image projected onto the camera, saving time, network resources, and memory space. For the LSTCAM, this *L1* delay is selected to be around 4 μs, to ensure that the *camera* trigger reaches every FEB just before the samples containing the shower image are overwritten. This further minimizes the dead-time of the DRS4 chip chain.

The *L1* signal is delayed at the TIB in a programmable Asynchronous Delay Line (ADL) composed of two chips: a 3D3428-15, which allows setting a delay of up to 255 steps of 15 ns, and a 3D3428-0.25, which can add 255 steps of 0.25 ns [28]. With the two lines in cascade, a maximum delay of 3888.75 ns with 0.25 ns precision is achieved. Other options to implement the ADLs were considered, like the FPGA IODELAY primitives [29]. However, they are not able to provide the

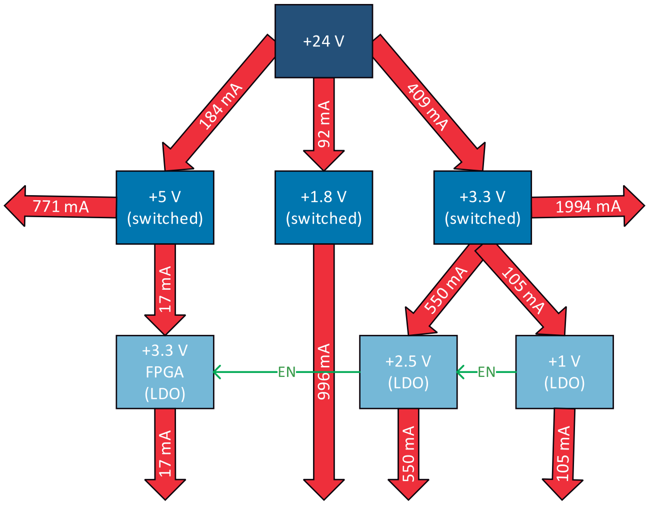


Fig. 7. Power supply scheme. All the power supplies were designed ad hoc with discrete components.

long delays required, so IODELAYs were discarded. After calibration in the first LST, the delay was adjusted to 3539 ns [30].

The fine resolution and low jitter provided by the delay chips are needed to preserve the timing accuracy of the *camera* trigger leading edge at the 1 ns level. With the selected ADLs, jitter in the *camera* trigger signal lower than 1 ns RMS was achieved, with respect to the *L1* input (see Fig. 6). It is important to point out that this number not only includes the jitter added by the ADLs, but also the one added by the logic operations in the FPGA. These logic operations will be explained in Section 7.

Finally, two other similar ADL circuits are used to delay the *Calibration* and *SinglePhe* triggers, because each one of these *trigger types* also needs the *camera* trigger to reach the FEBs at a fixed and specific time after the detection of the corresponding trigger input. These three trigger types can be interleaved during the data taking.

3.6. Power supplies

The TIB is powered from a +24 V power supply which is available for most parts of the camera. Then, from the 24 V, the TIB generates 6 internal power rails according to the scheme of Fig. 7. Switched power supplies were used for the power rails with higher power consumption (+5 V, +1.8 V, and +3.3 V), while linear regulators were preferred for those with low current consumption (+3.3V-FPGA, +2.5 V, and +1 V). It is important to take special care with the power sequencing. The FPGA requires receiving the +1 V and +1.8 V first, and later the +2.5 V and +3.3 V. On the other hand, the microcomputer requires to receive its lower voltage (1.8 V) at the same time as or after its high voltage (+3.3 V). The scheme of Fig. 7 was designed to meet these requirements: the computer will see +1.8 V and +3.3 V at nearly the same time, while the sequence for the FPGA will be:

1. +1.8 V, directly from the switched power supply.
2. +1 V, once the +3.3 V from the switched power supply will be available to feed the +1 V regulator.
3. +2.5 V, after receiving the *Power Good* signal from the +1 V power supply in the *Enable* input of the +2.5 V regulator.
4. +3.3V-FPGA, after the +3.3 V linear regulator receives the *Power Good* output from the +2.5 V regulator as *Enable* input.

It is important to note that the +3.3 V independent power rail is necessary to fulfill the power sequence and that a low noise power supply, like the one which is produced by a linear regulator, is recommended by the FPGA manufacturer.

4. Firmware architecture

Most of the functionalities implemented by the TIB are expressed in the logic defined in the firmware. Fig. 8 shows a simplified block diagram of the firmware with the main modules grouped by colors. The red blocks implement the *camera* trigger generation, essentially performing the OR of the enabled trigger signals from different origins. The yellow blocks extract the *trigger type* information, build the message, and send it to the UCTS and the microcomputer through two SPI interfaces. The blue boxes are in charge of the hardware stereo logic, contributing with the *Stereo* trigger signal to the *camera* trigger generation, and with the *stereo pattern* to the construction of the message sent to the UCTS and EVB. The gray blocks handle the whole camera busy state processing, which is rather complex and will be explained in detail in Section 8. The pink block on the left-hand side represents a *Pedestal* trigger generator, and the purple one refers to the clock generation, already commented in Section 3.2. Finally, the green blocks are the ones corresponding to the slow control and monitoring. The slow control block communicates with the microcomputer through an SPI link, which is used to read the trigger rates, configure the different blocks, manage the state machine, etc. In fact, there are control lines from the slow control block to nearly every other firmware module, but they have not been represented in Fig. 8 for the sake of clarity.

It is worth mentioning that the firmware is not loaded from an EEPROM, but the microcomputer loads the firmware stored in a file through an SPI interface at power-up. In this way, a firmware update is as simple as transferring a file to a computer and rebooting.

5. Software architecture

The computer uses a Raspbian Linux distribution to run the high-level software programs required. Essentially these programs are:

5.1. Slow control

The slow control software consists of an OPC-UA server [31] that exposes methods and parameters (datapoints in OPC-UA terminology) to CC, acting as the client. The protocol works over TCP/IP and the CC client can run the methods or perform a "set" or a "get" of each datapoint, covering all the control and monitoring requirements. The implementation of the OPC-UA server was a standard version provided by the CTA Consortium [32]. The specific work carried out for the TIB consisted in the development of a plugin in C++, containing the translation from the high-level commands received by the server, to the low-level actions required by the components (typically, to send and/or receive SPI messages).

The TIB OPC-UA server exposes 93 datapoints and 12 different methods. The configurable datapoints can be set independently or by running a method that receives an XML file with all the parameters and configures them. Regarding the monitoring datapoints (the ones which only admit a "get"), it is interesting to mention that 18 of them are refreshed automatically every few seconds. As the readings require taking control of SPI interfaces, they can cause concurrency problems that have to be solved with locks.

5.2. Data transmission to the EVB

The TIB delivers to the EVB a *camera event* fragment for each *camera* trigger, consisting of an *Event counter*, a redundant coarse timestamp (1 ms accuracy, 100 ns precision), the *trigger type*, and the *stereo* bit pattern. The data transmission from the TIB to the EVB follows closely the steps described in the TIB state machine (Section 9).

Before starting the data-taking, CC runs an OPC-UA method that launches the software handling the data transmission to the EVB (hereinafter DTX). In order to do so, the port and the IP address of the EVB should be provided to the DTX. Then, the DTX (client) opens a TCP/IP connection with the EVB (server) and then it gets suspended.

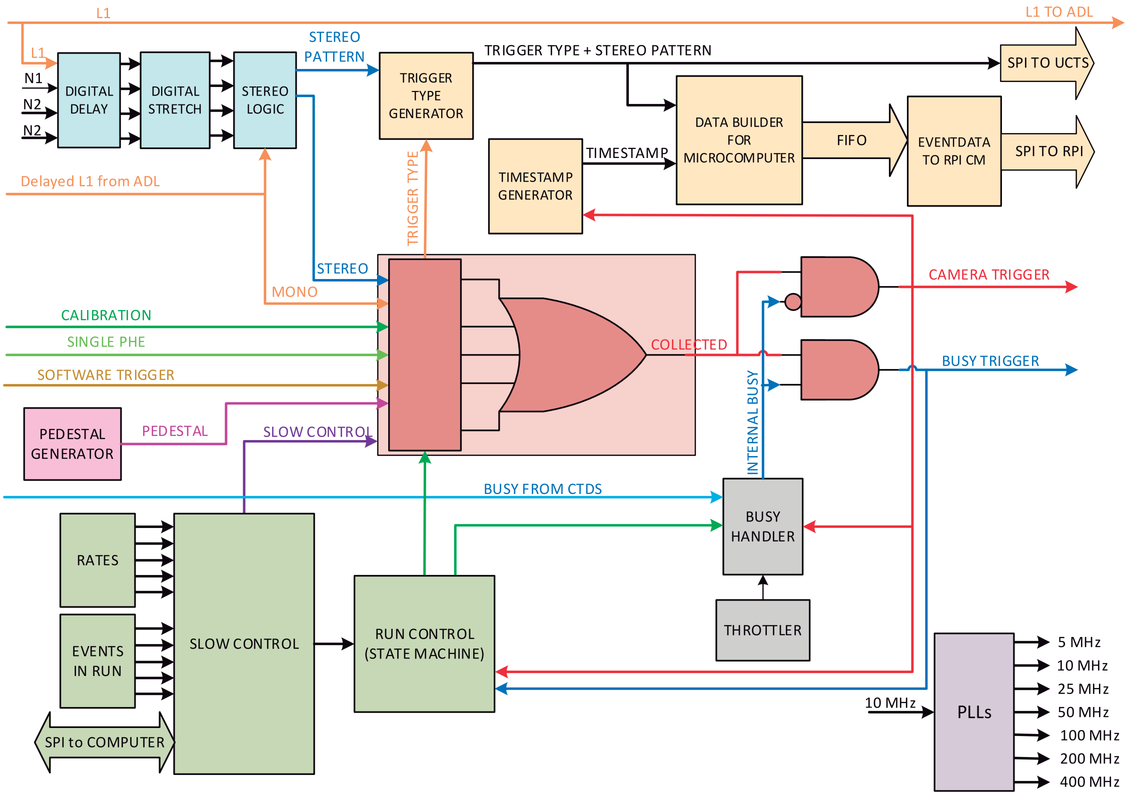


Fig. 8. Simplified scheme of the firmware architecture.

Table 1
Data structure for each event sent from the FPGA to the computer.

Bit	95 to 64	63 to 48	47 to 24	23 to 17	16 to 8	7	6 to 0
Field	Event counter	PPS counter	10 MHz counter	Zeros	Stereo pattern	busy	trigger type

When the first PPS arrives at the TIB FPGA from the UCTS, indicating the beginning of the run (see Section 9), the voltage of an FPGA output pin rises. The computer detects the voltage change in the pin, causing an interruption configured employing the library PiGPIO [33]. The DTX wakes up and obtains the UTC time for the first received PPS with NTP precision, typically around 1 ms [34]. Right afterward, the TIB starts delivering *camera* triggers to the FEBs, signaling the beginning of a data-taking run. For each *camera* trigger, the FPGA prepares a data packet consisting of the *Event counter*, the *PPS*, and *10* MHz counters, required to build the *TIB timestamps*, as well as the above-mentioned *trigger type*, and *stereo* bit patterns. These fields fill the structure shown in Table 1.

With the aim to get a more efficient transmission, the FPGA does not send every packet immediately but waits until accumulating a batch of 50 events (i.e. 600 bytes). When the FPGA batch is ready, the voltage level of a dedicated FPGA pin rises, and the computer can detect it. The DTX uses the PiGPIO library again to generate an interruption associated with this pin set, creating a thread that performs the following tasks:

1. it reads the 600 bytes through the SPI interface
2. it opens the data packet and includes a new field for each event with the absolute timestamp, calculated as the UTC of the first PPS, plus the number in the PPS counter field.
3. it rebuilds the 50-event packet with the new data and sends it through the TCP/IP interface to the EVB.

The DTX has strict real-time constraints. According to the LSTCAM specification, the *camera* triggers can happen at rates up to 15 kHz following a Poissonian distribution, while the NectarCAM specification corresponds to 7 kHz. The FIFO queue implemented in the firmware (see Fig. 8) buffers the bursts of events but, if the average event rate is higher than the maximum transfer rate of the system, the FIFO overflows. Then, some events in the corresponding 50-event packet are lost and the following 50-event packet becomes corrupted. After optimizing the DTX transmission parameters, the tests show that this takes place for event rates higher than 45 kHz. Nevertheless, if the TCP/IP connection is accidentally broken, the FIFO will overflow in a few seconds. In this case, the DTX keeps reading data from the SPI interface, disregarding the events that cannot be sent to the EVB. In the meantime, the DTX tries to reconnect. Once the reconnection succeeds, the new data reach the EVB normally [35]. The EVB is building the camera event according to the *Event* number of each camera fragment. Thus, if an event is found without the corresponding TIB information, it is flagged for debugging purposes, as it cannot be used for physics analysis.

5.3. FPGA configuration

This piece of software is a Python script that loads the firmware file into the FPGA using the GPIOs and an SPI interface. It is started automatically after booting, but it can also be launched by running a dedicated OPC-UA method.

6. The trigger types

At this point, it is useful to describe the characteristics of the different *trigger types* that can be processed by the TIB. All of them can be independently enabled or disabled by setting their corresponding OPC-UA datapoint. During the normal camera operation, only some of them are enabled at the same time, depending on the specific telescope use cases.

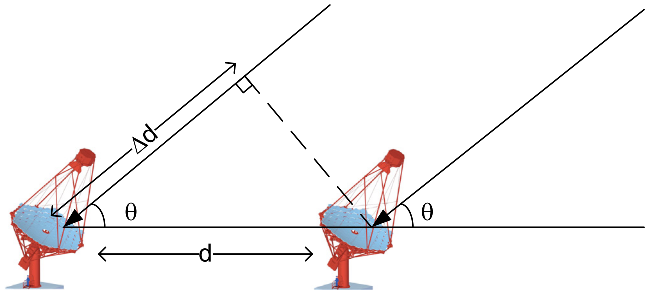
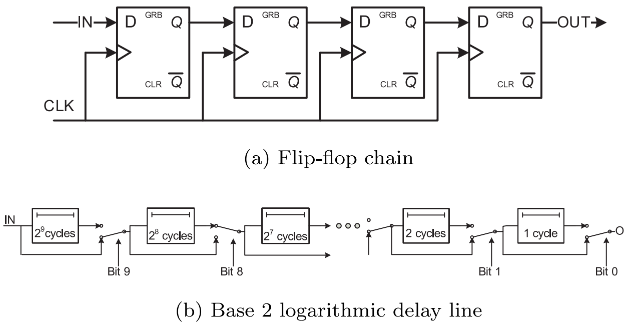


Fig. 9. Additional distance traveled by the light depending on the elevation angle θ.



(a) Flip-flop chain

(b) Base 2 logarithmic delay line

Fig. 10. Synchronous delay lines in the FPGA.

6.1. Mono

When the Mono trigger is enabled, the telescope is not operating in hardware stereo trigger mode and the *L1* trigger inputs can cause *camera* triggers to readout the camera. As it was explained in Section 3.5, the *L1* signals need to be delayed so the *camera* triggers reach the FEBs with the correct latency. This is the normal operation mode of NectarCAM.

6.2. Stereo

When the Stereo trigger is enabled, the Mono trigger is disabled. In this case, only the local *L1* triggers in coincidence with the *L1* signals from the neighbor LSTs causes *camera* triggers. This is the normal operation mode of LSTs. The distribution of the local *L1* trigger signals to the neighbors was already described in Section 3.4. Special care has to be taken by the TIB in this operating mode because the *L1* trigger pulse sent by a neighbor will take some time to reach the TIB of the local telescope through the optical fiber. This latency will depend on the length of the fiber and will be around 3 μs for the 2 most separated LSTs of the 4-LST grid. Moreover, depending on the pointing direction, the distance traveled by the Cherenkov photons will be different, as shown in Fig. 9. Again for the two most separated LSTs, this distance difference converts into a time difference of up to 560 ns (for a 20-degree elevation angle observation), which will be very relevant for the expected time coincidences among telescopes images in the range of 50 to 100 ns [36].

Thus, the first step before searching for stereoscopic detections of the same shower is to compensate for these delay differences. For a typical 50 ns time window, the low jitter feature of the asynchronous delay lines described in Section 3.5 is not required and we can implement the delays in the FPGA. The local *L1* and the *L1* signals from the neighbors feed chains of flip-flops with a 400 MHz common clock. The maximum delay can be close to 4 μs for the local *L1*, which is not delayed in any fiber, which means a 1600 flip-flop chain. In order to implement the adjustment capabilities, the flip-flops are arranged in blocks of 2^n units. The FPGA receives from the slow control the number of 2.5 ns cycles that should be delayed. So, if the bits coding the number are used to connect or bridge each 2^n flip-flop chain as shown in Fig. 10(b), a delay line with the latency indicated by the received number is obtained. Other alternatives to implement the delays were considered but finally discarded. For instance, a construction based on counters is not possible, because we would need a 12 bits counter able to update all the bits in 2,5 ns. The carry propagation takes more time for such a counter discarding this option.

Once the delays have been corrected, the output from each delay line is shaped to be as wide as the coincidence window. Then, all the lines with the delayed and stretched trigger pulses are sampled every 2.5 ns and the number of lines at a high level is counted. If this number is equal to or higher than the programmed stereo condition, a *Stereo* trigger takes place. This scheme is very flexible, allowing to change the coincidence window and the stereo condition through slow control, and providing simple scalability if more telescopes are integrated into the stereo trigger scheme.

However, it is important to realize that this algorithm checks if the stereo condition is fulfilled, but it does not provide a pulse with the correct timing to read the camera. Instead, it validates the local *L1*, acting as an "armed" signal, so the logic will accept the asynchronously delayed *L1* signal as a *Stereo* trigger.

6.3. Calibration

Calibration triggers take place when the camera is illuminated with a homogeneous and calibrated light intensity produced by the Calibration box [13]. When the Calibration box flashes, it sends a *Calibration* trigger pulse through an optical fiber to the TIB. Then, the TIB adds an adjusted delay with a dedicated ADL like the one described in Section 3.5 and uses the delayed *Calibration* trigger to generate a *camera* trigger that reaches the FEBs at the proper time.

6.4. Single Photoelectron

Single Photoelectron events (SinglePhe) take place when the entire camera, or only some pixels, are illuminated by a very faint source, so it can be assumed that each photosensor is detecting at most 1 photon. This kind of event is very useful to calibrate the gain of each pixel [14]. However, illuminating with such a faint light means that the camera must be closed, and the light is emitted by a special SinglePhe emitter. Similarly to the *Calibration* triggers, when the SinglePhe box flashes, it emits a *SinglePhe* trigger pulse that is sent to the TIB via the RJ45 interface. The TIB sends the trigger pulse through a different adjustable ADL, and the delayed *SinglePhe* trigger is used to generate a *camera* trigger synchronous with the flash.

6.5. Pedestal

Pedestal triggers are also used for telescope calibration and, by definition, take place at times not correlated with *L1* trigger or any other light source. The TIB can generate the pedestal triggers both following periodic and Poissonian random distributions, the latter aiming to mimic the natural NSB noise events recorded by the telescope. Poissonian random trigger generation is implemented with a 32 bits linear shift register (LSFR), as sketched in Fig. 11. When the generator is reset, a seed is loaded into the registers. Then, with every cycle of frequency 10 MHz/N, a pseudo-random number is generated. When the 9 LSB of this number in the registers match with a predefined combination, a *Pedestal* trigger is generated. The *Pedestal* trigger rate can be controlled by changing the divider parameter N. So, for $N = 1$ an average pedestal rate of ca. $\frac{10\text{ MHz}}{2^9} = 19.53$ kHz is obtained.

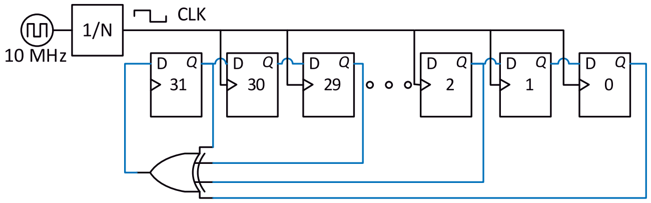


Fig. 11. Poissonian random number generator based on LSFR.

6.6. Scheduled

Sometimes it might be interesting to trigger several or all the telescopes of CTA at the same time. This can be done using the features of the White Rabbit technology. As every UCTS knows the UTC with nanosecond accuracy, it is possible to program a trigger for a specific time, like in an alarm clock. The trigger pulse reaches the TIB from the UCTS through a dedicated line in the RJ45 interface and, if enabled, it causes a *camera* trigger immediately.

6.7. Slow control

During the test and commissioning, it is interesting to be able to trigger the camera with a slow control command. In this case, the microcomputer sends the corresponding SPI message to the FPGA, which decodes it and generates a *camera* trigger. As the generation of these triggers involves a computer with an OS running a software program, it does not take place at a deterministic time.

6.8. Trigger type acquisition

Some of the trigger types described above are mutually exclusive, but others can be interleaved in the same run. It is very important to identify the *trigger type* of each event correctly because, depending on it, the SWAT and the EVB apply different algorithms to the data. As the trigger sources are independent, a single event might be caused by several trigger types and labeled accordingly.

The *trigger type* is obtained by taking a few samples of the trigger input lines, every time a leading edge of the *Collected* signal takes place (see Fig. 8). Every sample contains the status of the 7 trigger sources. 4 samples are taken with the 200 MHz clock, and then the samples are ORed. Therefore, every line in a high state during the sampled 20 ns will be included in the *trigger type*. Finally, the *trigger type* is sent to the UCTS and the microcomputer following the data flow represented with the yellow boxes of Fig. 8.

7. The trigger path

As it was outlined in Section 3.5, a very low jitter (≤ 1 ns) in the leading edge of the *camera* trigger pulses with respect to the observed light signal is required by LSTCAM and NectarCAM. Since the Cherenkov showers can happen at any time, the trigger signal cannot be synchronous to any clock. Instead, it must follow a fully asynchronous path, with a constant latency:

- In the LSTCAM Level 0, adjustable analog delays are calibrated in each pixel, with the aim to compensate for the distinct transit times due to different high voltage levels in each PMT [37]. In NectarCAM the compensation is achieved by means of adjustable digital delays implemented in the backplanes connected to each FEB.
- In the TCDS, additional programmable analog delays are added to the *L1* trigger signals generated in each FEB to equalize the time required to reach the TIB. Thus, the TIB receives the *L1* around 100 ns after the light excess appears in the pixels at any place in the camera. The same strategy is applied to the distribution of the *camera* trigger down to the FEBs [10].
- In the TIB, the handling of the trigger pulses in the FPGA is done with combinational logic, and the additional delays are added employing ADLs (Section 3.5).

Apart from the mono trigger type, other trigger types like *Calibration* or *SinglePhe* also require the low jitter characteristic to provide the *camera* trigger well synchronized with the corresponding light source.

In particular, as described in Fig. 8, the process applied to the asynchronous trigger inputs consists of:

1. Mask the trigger inputs to enable or disable them according to the slow control datapoint. The mask must be configured before triggering starts, to avoid partial trigger pulses.
2. Perform the logic OR of the trigger inputs. The output is the *Collected* signal in Fig. 8.
3. The *Collected* signal is delivered to two AND gates with the *busy* and the *inverted busy* signal, giving rise to a *camera* trigger or a *busy* trigger.

8. Handling of the camera busy state

When a *camera* trigger is distributed, the FEBs start digitizing the samples and cannot accept new triggers until they have finished. However, the *L1* triggers keep arriving at the TIB during this dead time, and they must be further processed by the TIB. Even if these triggers are not associated with camera events, their timestamps and *trigger type* can be useful for the SWAT, which could confirm Cherenkov showers observed by other telescopes. With this idea in mind, the *L1* triggers are not just discarded, but they are sent to the UCTS through a dedicated line in the RJ45 interface for the so-called *busy triggers*.

8.1. Busy signals

The TCDS provides a *busy* signal to the TIB, generated as the OR of the busy state of every FEB. However, this *busy* signal needs some time to propagate and reach the TIB (ca. 100 ns). During this time, if an additional *L1* trigger reaches the TIB, it should be identified as a *busy* trigger and not as a *camera* trigger. With this aim, the *busy handler* module of Fig. 8 works out an *internal busy* signal, which is set high with the falling edge of the *camera* trigger. Then, the *internal busy* maintains its high state during a minimum time (500 ns), longer than the busy propagation time through the TCDS, but shorter than the dead time. After this minimum time, the *internal busy* remains high until the *busy* signal from the TCDS changes to 0. A low level in the *busy* input guarantees that every FEB is ready to receive a new *camera* trigger and digitize a new event. Nevertheless, the *internal busy* cannot be lowered immediately, but only after checking that no *busy trigger* is being delivered to the UCTS. This check is required to avoid splitting the *Collected* pulse into two pulses: A *busy* trigger first and a *camera* trigger afterward.

8.2. The Throttler

The dead time of the LSTCAM and NectarCAM is around 7 μs per event.[1] However, the fact that two consecutive *camera* trigger events separated by more than 7 μs can be processed does not mean that the telescope can process data at a sustainable rate of $\frac{1}{7\ \mu\text{s}} \approx 143$ kHz.

[1] A new version of Nectar chip is expected to lower the dead time by an order of magnitude.

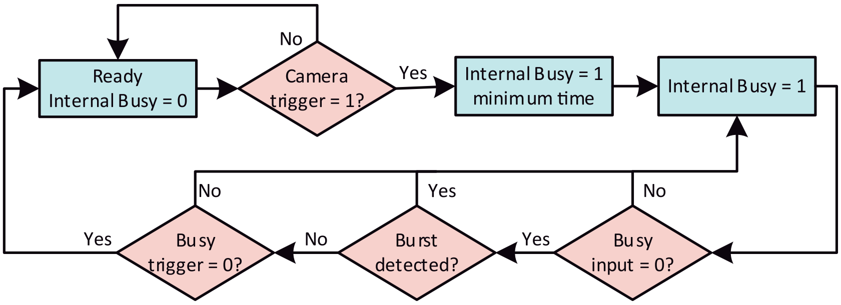


Fig. 12. TIB busy handling scheme flowchart.

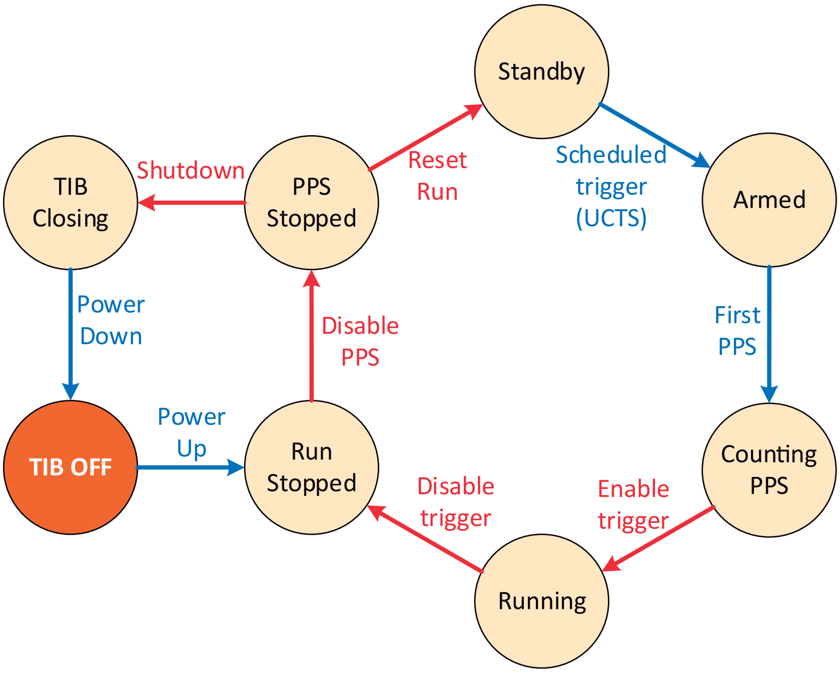


Fig. 13. TIB state machine.

The LSTCAM is designed to operate at 15 kHz and the NectarCAM at 7.5 kHz. Therefore, if a burst of triggers with a short separation appears, the excess of data would be buffered in the camera data switches and the EVB [38]. However, if the burst contains too many triggers, the buffer of the data switches might overflow losing data packets from random FEBs. To avoid that, the TIB contains a Trigger Rate Cap mechanism in the form of a firmware module labeled as Throttler in Fig. 8.

The trigger rate cap can be written as “no more than 30 *camera* triggers in 1 ms”. The Throttler uses the *camera* trigger signal to feed a synchronous delay line like the one shown in Fig. 10(a), with a length of 1 ms and driven by a 5 MHz clock in order to avoid a too long flip-flop chain. A counter adds 1 to its account every time a new trigger pulse enters into the delay line, and subtracts 1 every time a trigger pulse comes out. In this way, one can always determine the number of triggers that have taken place during the last ms. If the number exceeds the configured value, the *internal busy* will not be lowered. Thus, the complete flowchart of the TIB busy handling scheme is as shown in Fig. 12.

8.3. *Processing of camera and busy trigger information*

The information associated with each trigger signal that is generated by the TIB is processed differently depending on whether it corresponds to a *busy* trigger or a *camera* trigger.

8.3.1. *The camera triggers*

- The *camera* trigger pulses are sent to the UCTS and the TCDS.
- The *trigger type* + *stereo pattern* is sent to the UCTS through an SPI link right after each trigger pulse.
- The *trigger type* + *stereo pattern* + *TIB timestamp* is sent to the microcomputer and, in turn, to the EVB with the DTX described in Section 5.2.

8.3.2. *The busy triggers*

- The *busy* trigger pulses are sent only to the UCTS.
- The *trigger type* + *stereo pattern* is only sent to the UCTS through an SPI link right after each trigger pulse.
- The *TIB timestamp* is not generated. There is no data corresponding to this event delivered to the microcomputer.

9. The state machine

The final item in the TIB architecture is its state machine, which plays a central role in the whole camera data-taking scheme.

Looking for redundancy, all events recorded by the camera are identified by their event numbers and their timestamps. The data packets produced by the 265 FEBs also contain these two parameters, which are used by the EVB. In order to maintain coherence among these numbers, it is critical to use a common starting time for all the systems (FEBs, TIB, and UCTS) in a data-taking run. Moreover, it is interesting to have a common starting time available for every telescope of CTA. The perfect candidate to mark the start of the run is the PPS signal, as it is distributed by the White Rabbit network to each telescope with nanosecond accuracy and, in turn, it is distributed by the TCDS to every FEB isochronously. The TIB state machine, represented in Fig. 13, was developed with the aim to use the PPS to mark the beginning of the run and to allow time for slow control operations, such as resetting counters or reading statistics.

After power-up, the TIB enters in state *Run Stopped*. In this state, the TIB delivers the PPS signal to the TCDS (required by the LSTCAM FEBs for synchronization), but not *camera* or *busy* triggers. By running the slow control command *Disable PPS*, the PPS signal stops being distributed. In the state *PPS Stopped*, all the counters are reset by CC in every FEB, including Events, 10 MHz, and PPS counters. This operation takes typically 30 s. Then, the command *Reset Run* resets the above-mentioned counters in the TIB, which still distributes neither PPS nor triggers. The TIB reaches the *Standby* state, with all the FEBs ready to receive PPS and *camera* trigger.

Just after the PPS previous to the first PPS signaling the start of the run, a *Scheduled* trigger is sent to the TIB from the UCTS. In *Standby*, this signal is considered as a herald, indicating that the next PPS will be the first one of the run, changing the TIB state to *Armed*. Using a *Scheduled* trigger has the advantage that it can be scheduled in many telescopes to happen at the same time. The UCTS is delivering this signal to the TIB, which has almost 1 s to get ready to receive it.

Then, the first PPS of the run arrives at the TIB, marking the start of its PPS and 10 MHz counters which are used to build the *TIB timestamp*. This first PPS is also distributed by the TIB to the TCDS, reaching all the FEBs isochronously, so all the FEBs start incrementing their PPS and 10 MHz counters in a synchronized way. In this situation, incrementing time counters but not distributing triggers yet, the TIB is in state *Counting PPS*.

Finally, with the command *Enable trigger* the TIB starts delivering *camera* and *busy* triggers, as well as their corresponding *trigger types*, *stereo patterns*, and *TIB timestamps*. As all the triggers have to go through the TIB, start triggering at a specific time is not critical for data coherence. The first event is the first one for the TIB, the FEBs, and the UCTS, independently of when it happens. And it is identified by the same counter in the TIB, the FEBs, and the UCTS, as well as absolutely timestamped by the UCTS and the TIB.

When the run is over, the command *Disable trigger* is executed, and the TIB stops delivering *camera* triggers to the FEBs, reaching the *Run Stopped* state. The event counters are not increased anymore and there is no more data sent to the EVB or the UCTS. However, the run is not finished yet, because the PPS is still being distributed and the time counters did not stop. The run is formally finished in state *PPS Stopped*. In this state, the TIB updates its datapoints with statistics of the run, such as the number of triggers of each type or the number of seconds that the previous run lasted.

To switch off the TIB, the command *Shutdown* must be called from the state *PPS Stopped*. The TIB is in *TIB Closing* state for a few seconds while shutting down the microcomputer. Then, the TIB is ready for power off.

10. Conclusion

A Trigger Interface Board has been developed for the LSTCAM and NectarCAM of CTA. It is a core component of the telescope cameras because it is in charge of managing the different camera trigger and clock signals. Among its features, one can highlight the generation of the *camera* trigger signal (actually signaling the camera read-out), the whole camera busy handling mechanism, the construction of the *event class* and *timestamp* information, and the implementation of the LST hardware stereo trigger scheme. It is currently at the end of its commissioning phase, along with the rest of the LST and NectarCAM elements [6,39]. The system is based on an FPGA and a microcomputer, combining the ability to handle fast digital signals with high-level software features, like an OPC-UA server or TCP-IP communication. The multi-level design (hardware–firmware–software) provides great flexibility to develop new features or modify the behavior of the current ones while keeping the same hardware.

CRediT authorship contribution statement

Luis A. Tejedor: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Writing – review & editing, Supervision. **Juan A. Barrio:** Project administration, Funding acquisition, Resources. **Pablo Peñil:** Methodology, Software, Validation, Data curation. **Alejandro Pérez:** Software, Validation, Investigation, Visualization. **Diego Herranz:** Conceptualization, Methodology, Investigation. **Jorge Martín:** Software, Visualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We acknowledge the support of the PID2019-104114RB-C32 and FPA2017-82729-C6-3-R grants of the Agencia Estatal de Investigación del Ministerio de Ciencia, Innovación y Universidades. We would also like to gratefully acknowledge the support from the CTA collaboration, especially from the LSTCAM and NectarCAM teams.

References

[1] Cherenkov telescope array site, URL http://www.cta-observatory.org.

[2] A. Hillas, Evolution of ground-based gamma-ray astronomy from the early days to the cherenkov telescope arrays, Astropart. Phys. 43 (2013) 19–43, http://dx.doi.org/10.1016/j.astropartphys.2012.06.002.

[3] B. Acharya, M. Actis, T. Aghajani, et al., Introducing the CTA concept, Astropart. Phys. 43 (2013) 3–18, http://dx.doi.org/10.1016/j.astropartphys.2013.01.007.

[4] P.P. nil, L.A. Tejedor, J.A. Barrio, M. López, A trigger interface board to manage trigger and timing signals in cta large-sized telescope and medium-sized telescope cameras, in: Proceedings of the 35th International Cosmic Ray Conference (ICRC2017), Busan, Korea, 2017, arXiv:1709.04692.

[5] L.A. Tejedor, A new analog trigger system for the cherenkov telescope array, (Ph.D. thesis), Polytechnic University of Madrid, School of Telecommunication Engineering, 2014, URL http://oa.upm.es/32941/.

[6] J. Barrio, Status of the large size telescopes and medium size telescopes for the cherenkov telescope array observatory, Nucl. Instrum. Methods Phys. Res. A 952 (2020) 161588, http://dx.doi.org/10.1016/j.nima.2018.11.047, 10th International Workshop on Ring Imaging Cherenkov Detectors (RICH 2018).

[7] J.-F. Glicenstein, M. Shayduk, the NectarCAM Collaboration, the CTA Consortium, NectarCAM, a camera for the medium sized telescopes of the cherenkov telescope array, in: AIP Conference Proceedings 1792, 080009 (2017), AIP Publishing, 2017, http://dx.doi.org/10.1063/1.4969030.

[8] S. Ritt, R. Dinapoli, U. Hartmann, Application of the DRS chip for fast waveform digitizing, Nucl. Instrum. Methods Phys. Res. A 623 (1) (2010) 486–488, http://dx.doi.org/10.1016/j.nima.2010.03.045.

[9] C. Naumann, E. Delagnes, J. Bolmont, P. Corona, D. Dzahini, F. Feinstein, D. Gascón, J.-F. Glicenstein, F. Guilloux, P. Nayman, F. Rarbi, A. Sanuy, J.-P. Tavernet, F. Toussenel, P. Vincent, S. Vorobiov, New electronics for the cherenkov telescope array (NECTAr), Nucl. Instrum. Methods Phys. Res. A 695 (2012) 44–51, http://dx.doi.org/10.1016/j.nima.2011.11.008.

[10] L.A. Tejedor, M. Barceló, J. Boix, D. Herranz, G. Martínez, J.A. Barrio, O. Blanch Bigas, C. Delgado, R. López, An analog trigger system for atmospheric cherenkov telescope arrays, IEEE Trans. Nucl. Sci. 60 (3) (2013) 2367–2375, http://dx.doi.org/10.1109/TNS.2013.2257852.

[11] U. Schwanke, M. Shayduk, K.-H. Sulanke, S. Vorobiov, R. Wischnewski, A versatile digital camera trigger for telescopes in the cherenkov telescope array, Nucl. Instrum. Methods Phys. Res. A 782 (2015) 92–103, http://dx.doi.org/10.1016/j.nima.2015.01.096.

[12] I. Oya, E. Antolini, L. Baroncelli, J. Borkowski, A. Bulgarelli, A. Carosi, J. Colome, V. Conforti, M. Fuessling, C. Hoischen, J. Jacquemier, E. Lyard, G. Maurin, D. Melkumyan, K. Mosshammer, N. Parmiggiani, I. Sadeh, T. Schmidt, U. Schwanke, J. Schwarz, G. Tosti, R. Walter, P. Wegner, The array control and data acquisition system of the cherenkov telescope array, in: Proc. ICALEPCS'19, in: International Conference on Accelerator and Large Experimental Physics Control Systems, (17) JACoW Publishing, Geneva, Switzerland, 2020, pp. 1046–1051, URL https://jacow.org/icalepcs2019/papers/wempr005.pdf.

[13] M. Palatiello, M. Iori, F. Cassol, D. Cauz, F. Ferrarotto, Performance of the INFN camera calibration device of the first large size telescope in the cherenkov telescope array, in: Proc. 36th International Cosmic Ray Conference (ICRC2019), Madison, Wisconsin, USA, 2019, URL https://arxiv.org/abs/1909.08475.

[14] B. Biasuzzi, K. Pressard, J. Biteau, B. Geoffroy, C.D. Goncalves, G. Hull, M. Imre, M. Josselin, A. Maroni, B. Mathon, L. Seminor, T. Suomijarvi, T.N. Trung, L. Vatrinet, P. Brun, S. Caroff, S. Fegan, O. Ferreira, P. Jean, S. Karkar, J.-F. Olive, S. Rivoire, P. Sizun, F. Thiant, A. Tsiahina, F. Toussenel, G. Vasileiadis, Design and characterization of a single photoelectron calibration system for the NectarCAM camera of the medium-sized telescopes of the cherenkov telescope array, Nucl. Instrum. Methods Phys. Res. A 950 (2020) 162949, http://dx.doi.org/10.1016/j.nima.2019.162949.

[15] The White Rabbit Project, Open Hardware Repository, 2020, URL https://ohwr.org/projects/white-rabbit.

[16] J. Sanchez-Garrido, A. Jurado, M. Jimenez-Lopez, A.K. Balzer, H. Prokoph, M. Stephan, D. Berge, M. Rodriguez-Alvarez, J. Diaz, A white rabbit-synchronized accurate time-stamping solution for the small-sized cameras of the cherenkov telescope array, IEEE Trans. Instrum. Meas. (2020) 1, http://dx.doi.org/10.1109/TIM.2020.3013343.

[17] C. Champion, S. Colonges, Y. Moudden, R. Oger, M. Punch, Ticks: A flexible white-rabbit based time-stamping board, in: Proc. of International Conference on Accelerator and Large Experimental Control Systems (ICALEPCS'17), (16) JACoW, Barcelona, Spain, 2017, pp. 622–626, http://dx.doi.org/10.18429/JACoW-ICALEPCS2017-TUPHA090.

[18] D. Hoffmann, J. Houles, Prototyping a 10 gigabit-ethernet event-builder for the CTA camera server, J. Phys. Conf. Ser. 396 (1) (2012) 012024, http://dx.doi.org/10.1088/1742-6596/396/1/012024.

[19] Compute Module 3, The Raspberry Pi Foundation, 2020, URL https://www.raspberrypi.org/products/compute-module-3/.

[20] Artix 7, Xilinx Inc., 2020, URL https://www.xilinx.com/products/silicon-devices/fpga/artix-7.html.

[21] IEC 61754-20:2012 Fibre Optic Interconnecting Devices and Passive Components - Fibre Optic Connector Interfaces - Part 20: Type LC Connector Family, International Electrotechnical Comission, 2012, URL https://webstore.iec.ch/publication/5827.

[22] TMC-5A40-008 High Speed VCSEL TO-46 Metal Can, TrueLight Corp, 2020, URL https://www.truelight.com.tw/WebFiles/201810261540156692384716TMC-5A40-008.pdf.

[23] Si PiN Photodiode S5973, Hamamatsu Photonics, 2020, URL https://www.hamamatsu.com/eu/en/product/type/S5973/index.html.

[24] ISO/IEC 11801 Information Technology - Generic Cabling for Customer Premises.

[25] AD8009, 1 GHz, 5500 V/μs, Low Distortion Amplifier, Analog Devices Inc., 2020, URL https://www.analog.com/media/en/technical-documentation/data-sheets/AD8009.pdf.

[26] A. Sanuy, D. Gascon, J.M. Paredes, L. Garrido, M. Ribó, J. Sieiro, Wideband (500 MHz) 16 bit dynamic range current mode PreAmplifier for the CTA cameras (PACTA), J. Instrum. 7 (01) (2012) C01100, http://dx.doi.org/10.1088/1748-0221/7/01/c01100.

[27] Rail to Rail, Very Fast, 2.5V to 5.5V Single-Supply LVDS Comparators ADCMP604/ADCMP605, Analog Devices Inc., 2020, URL https://www.analog.com/media/en/technical-documentation/data-sheets/ADCMP604_605.pdf.

[28] Monolithic 8-bit programmable delay line (Series 3D3428 - Low Noise), Data Delay Devices Inc., 2020, URL https://0916d893-7fc9-4aeb-9ef1-94eb87658d15.filesusr.com/ugd/8d75f4_ad492441e22d4da0a625d2e28a7c7961.pdf.

[29] 7 Series FPGAs Select IO Resources, Xilinx Inc., 2018, URL https://www.xilinx.com/support/documentation/user_guides/ug471_7Series_SelectIO.pdf.

[30] S. Nozaki, K. Awai, A. Bamba, J.A. Barrio, M.I. Bernardos, O. Blanch, J. Boix, F. Cassol, Y. Choushi, C. Delgado, C. Diaz, N. Fouque, L. Freixas, P. Gliwny, S. Gunji, D. Hadasch, D. Hoffmann, J. Houles, Y. Inome, Y. Iwamura, L. Jouvin, H. Katagiri, K. Kawamura, D. Kerszberg, Y. Konno, H. Kubo, J. Kushida, Y. Kobayashi, R. López-Coto, G. Martinez, S. Masuda, D. Mazin, A. Moralejo, E. Moretti, T. Nagayoshi, T. Nakamori, K. Nishijima, Y. Nogami, L. Nogués, H. Ohoka, T. Oka, N. Okazaki, A. Okumura, R. Orito, J.-L. Panazol, R. Paoletti, C. Pio, M. Polo, J. Prast, T. Saito, S. Sakurai, J. Sitarek, Y. Sunada, M. Suzuki, M. Takahashi, K. Tamura, M. Tanaka, L.A. Tejedor, Y. Terada, M. Teshima, Y. Tsukamoto, T. Yamamoto, Calibration and performance of the readout system based on switched capacitor arrays for the large-sized telescope of the cherenkov telescope array, in: C.J. Evans, J.J. Bryant, K. Motohara (Eds.), Ground-Based and Airborne Instrumentation for Astronomy VIII, Vol. 11447, SPIE, International Society for Optics and Photonics, 2020, pp. 64–77, http://dx.doi.org/10.1117/12.2560018.

[31] Unified Architecture, The OPC Foundation, 2020, URL https://opcfoundation.org/about/opc-technologies/opc-ua/.

[32] I. Oya, B. Behera, E. Birsin, H. Koeppel, D. Melkumyan, T. Schmidt, U. Schwanke, P. Wegner, S. Wiesand, M. Winde, C. Consortium, Evaluating the control software for CTA in a medium size telescope prototype, J. Phys. Conf. Ser. 396 (1) (2012) 012037, http://dx.doi.org/10.1088/1742-6596/396/1/012037.

[33] The PiGPIO Library, abyz.me.uk, 2020, URL http://abyz.me.uk/rpi/pigpio/.

[34] J. Martín, Absolute Event-Time-Stamping Assignation and Delivery with a Raspberry Pi Compute Module and an FPGA (Bachellor thesis), Complutense University of Madrid, Physics Faculty, 2020.

[35] P.P. nil, Desarrollo, Pruebas y Mejoras de Firmware y Software para el Sistema de Trigger Estéreo del Cherenkov Telescope Array (Masther's thesis), Complutense University of Madrid, Physics Faculty, 2016.

[36] J. Aleksić, R. Paoletti, et al., for the MAGIC Collaboration, Performance of the MAGIC stereo system obtained with crab nebula data, Astropart. Phys. 35 (7) (2012) 435–448, http://dx.doi.org/10.1016/j.astropartphys.2011.11.007.

[37] L.A. Tejedor, J. Boix, F. Dubois, Y. Konno, D. Herranz, J.A. Barrio, J.I. Alonso, An analog delay compensation system to reduce the effect of variable transit time in PMTs, IEEE Trans. Nucl. Sci. 60 (4) (2013) 2905–2911, http://dx.doi.org/10.1109/TNS.2013.2271300.

[38] D. Hoffmann, J. Houles, P. Sizun, L. Frederic, Y. Moudden, J.-F. Glicenstein, G. W., B. Raydo, Stress testing ethernet switches for nectarcam in the cherenkov telescope array with a synchronous UDP frame generator, in: 34th International Cosmic Ray Conference (ICRC2015), in: Proceedings of Science (POS) ICRC 2015, vol. 697, The Hague, Netherlands, 2015, URL http://hal.in2p3.fr/in2p3-01431583.

[39] T. Tavernier, J.-F. Glicenstein, F. Brun, Status and performance results from NectarCAM – A camera for CTA medium sized telescopes, in: Proc. 36th International Cosmic Ray Conference (ICRC2019), 2019, URL https://arxiv.org/abs/1909.01969.