UNIVERSITÀ DEGLI STUDI DI PADOVA

Dipartimento di Fisica e Astronomia "Galileo Galilei"

Corso di Laurea Magistrale in Fisica

Tesi di Laurea

# Design and tests of the FPGA embedded trigger algorithms for the large PMT JUNO Electronics

Relatore

Prof. Alberto Garfagnini

Correlatore

Dr. Ing. Marco Bellato

Laureando

Filippo Marini

Anno Accademico 2017/2018

# Abstract

The JUNO Jiangmen Underground Neutrino Observatory, a 20 kton multi-purpose underground liquid scintillator detector, is under construction in the south of China. The main physics target of JUNO is the determination of the neutrino mass hierarchy, which will be accessible through the measurement of the antineutrino spectrum coming from two high power nuclear complexes located about 53 km away from the experimental site. An excellent energy resolution and a large JUNO detector fiducial volume are a key ingredients for addressing many important topics in neutrino and astro-particle physics.
After a brief description of the neutrino physics involved the experimental setup is characterized. The liquid scintillator detector is surrounded by 18'000 20-inch PMTs as well as 25'000 3-inch PMTs in order to reconstruct any interesting neutrino event.
The 20-inch PMTs readout electronics is the working environment of this thesis, with an important focus on the Global Control Unit, the intelligent component, a board mounting the FPGA that processes the raw data coming from the PMT. The aim of the thesis is to illustrate the design and the tests of the trigger implemented in the FPGA, used to discriminate the signal pulses from the noise. The designing of a triggering algorithm has been driven by the need for a discrimination of low-amplitude signals. The JUNO experiment requires an evaluation of even single photonelectron signals originated by a large PMT. In this conditions, the signal amplitude is comparable to the electronics background noise, making a simple leading-edge trigger not enough efficient.
After having defined all the steps of the algorithm, the data processing from the raw digitized PMT signal to the trigger output, on the MATLAB® simulations results are presented. Finally, the changes made to the algorithm during its implementation are highlighted, focusing on the improvements made for increasing the overall efficiency. Some implementation simulations are shown.
A testing of the trigger algorithm is also performed, in order to verify the successful implementation into the FPGA. Since the algorithm obtains the energy of the signal's pulses before triggering, evidence which shows its accuracy are presented. The purpose of second test performed is to illustrate its efficiency, comparing it to a simple leading-edge trigger algorithm.

# Contents

Chapter 1

# The Theory Behind The Experiment

Precise measurements of the $\theta_{13}$ neutrino oscillation parameter by the Daya Bay [1], Reno [2] and Double Chooz [3] experiments, have opened the path to the determination of the neutrino mass hierarchy. Indeed weather the $\nu_3$ neutrino mass eigenstate is heavier or lighter than the $\nu_1$ and $\nu_2$ mass eigenstates is one of the remaining undetermined fundamental aspects of the Standard Model in the lepton sector. [4] Mass hierarchy determination would have an impact in the quest of the neutrino nature (Dirac or Majorana mass terms) towards the formulation of a theory of flavour. Measuring the energy spectra of neutrinos coming from nuclear reactors at a medium distance is a clean experimental method to determine the mass hierarchy without exploring neutrinos matter effects [5]. The Jiangmen Underground Neutrino Observatory (JUNO) is a large liquid scintillator neutrino detector under construction in the south of China [?]. Thanks to the large 20 kton active mass and unprecedented energy resolution (3% at 1 MeV) it will allow to determine the neutrino mass hierarchy and to precisely measure the neutrino mixing parameters, $\theta_{12}$, $\Delta m^2_{12}$ and $\Delta m^2_{ee}$ below the 1% level.

## 1.1 Neutrino oscillations

The JUNO approach to neutrino mass hierarchy determination, is to detect neutrino oscillation in vacuum at a medium baseline from the source. The experiment will detect electron antineutrino interactions in the main detector thanks to the inverse beta decay reaction $\bar{\nu}_e + p \to e^+ + n$. The energy deposited by the positron annihilating in the liquid scintillator is strictly proportional to the energy of the incoming antineutrinos. The time coincidence between the positron interaction and the subsequent neutron capture on protons allows to identify efficiently the neutrino interaction, even in the presence of uncorrelated background. The electron antineutrino survival probability can be written as

$$P_{ee} = 1 - sin^2 2\theta_{13} \cdot (cos^2\theta_{12} sin^2\Delta_{31} + sin^2\theta_{12} sin^2\Delta_{32}) - sin^2 2\theta_{12} \cdot cos^4\theta_{13} sin^2\Delta_{12} \quad (1.1)$$

where $\Delta_{ij} = \Delta m^2_{ij} L/4E_\nu$.
With the approximation $\Delta m^2_{32} \approx \Delta m^2_{31}$, it is possible to rewrite the survival probability and make explicit the mass hierarchy dependence in the formula

$$\begin{aligned} P_{ee} = 1 - & cos^4\theta_{13}sin^2 2\theta_{12}sin^2\Delta_{21} - sin^2\theta_{13}sin^2|\Delta_{31}| \\ & - sin^2\theta_{12}sin^2 2\theta_{13}sin^2\Delta_{21}cos2|\Delta_{31}| \\ & \pm (sin^2\theta_{12}/2)sin^2 2\theta_{13}sin^2 2\Delta_{21}sin^2 2|\Delta_{31}| \end{aligned} \tag{1.2}$$

The sign flip in the last term is due to the neutrino mass hierarchy possibility: the direct hierarchy gives a positive contribution, while it is negative for the inverse mass ordering. The effect is small, but fortunately not negligible due to the relatively large value for $\theta_{13}$.

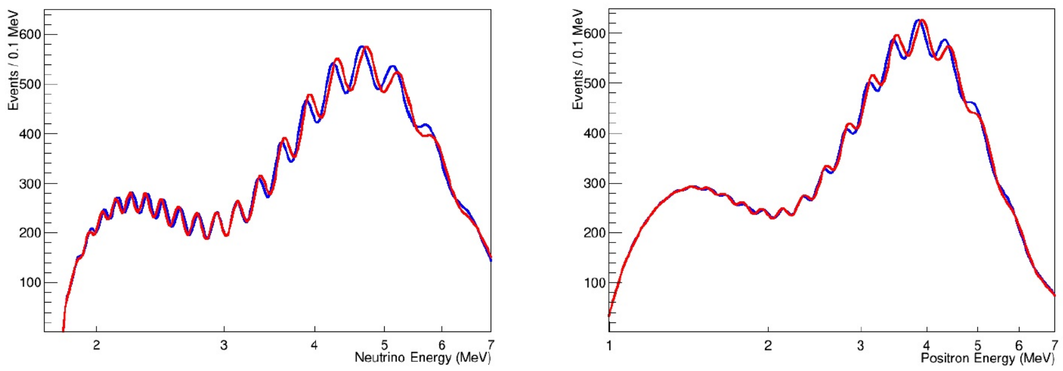


FIGURE 1.1: *Left: electron antineutrino energy spectrum at a baseline of about 53 km for an infinite energy resolution. Right: positron visible energy spectrum assuming a 3% at 1 MeV detector energy resolution. Both normal (red) and inverted (blue) hierarchies are shown in the plots.*

The effect of the neutrino mass hierarchy on the electron antineutrino energy spectrum, measured at a medium-baseline of 53 km, is shown in Fig. 1.1. Thanks to the infinite energy resolution assumed for the reconstructed antineutrino energy (left plot), the two curves are clearly distinguishable. The right plot of the same figure shows the reconstructed positron visible energy, having assumed a 3% energy resolution at 1 MeV. The finite resolution of detectors tends to wash away the characteristic features of the ripples; if the resolution deteriorates further the peaks and throats of the spectrum would start to disappear, making it very hard or almost impossible to unfold the neutrino mass hierarchy from the collected spectra. Detailed studies have shown that assuming a large scintillator detector with a 20 kton mass, a 3% energy resolution and an exposure time of 6 years provide a 4 $\sigma$ discrimination power [6]. Systematic effects would affect the result and worsen the discrimination power. Possible effects could come to the non exact baseline of the detector with respect to the reactor cores. Other important effects are the shape uncertainty of the reactor spectrum and the background discrimination [6].
Beyond mass hierarchy and precision determination of the neutrino oscillation parameters, a large liquid scintillator detector can provide fundamental results on many topics in astroparticle physics, like supernova burst and supernova diffuse neutrinos, solar neutrinos, atmospheric neutrinos, geo-neutrinos, nucleon decay, indirect dark matter searches and a number of additional exotic searches. A reference to the rich physics program of JUNO can be found here [6].

Chapter 2

# JUNO Experiment

JUNO [6] will detect reactor antineutrinos from Nuclear Power Plant (NPP). The mass hierarchy determination requires equal baselines from the detector to all reactor cores to avoid cancellation of the oscillation dephasing effect.
The site location is optimized to have the best sensitivity for the mass hierarchy determination, which is at 53 km from both the Yangjiang and Taishan NPPs. The neutrino detector is a liquid scintillator detector with a 20 kton fiducial mass, deployed in an underground laboratory about 700m deep. The experimental site and the detector will be described in the following.
The JUNO project was approved by Chinese Academy of Sciences in February 2013. Data taking is expected to start in 2020.

## 2.1 Experimental site

The JUNO experiment locates in Jinji town, Kaiping city, Jiangmen city, Guangdong province. The geographic location is East longitude $112°31'05"$ and North latitude $22°07'05"$. The experimental site is 43 km to the southwest of the Kaiping city, a county-level city in the prefecture-level city Jiangmen in Guangdong province. There are five big cities: Guangzhou, Hong Kong, Macau, Shenzhen, and Zhuhai, all in about 200 km drive distance, as shown in Fig. 2.1.

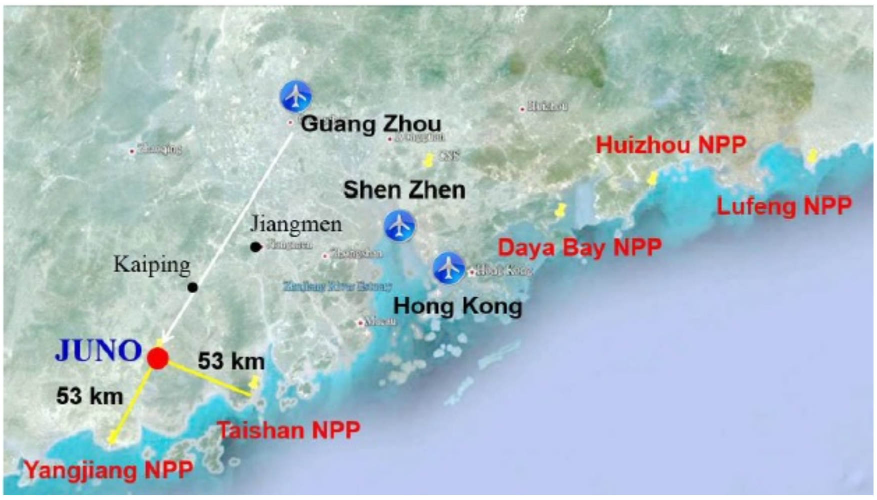


FIGURE 2.1: *Location of the JUNO site. The distances to the nearby Yangjiang NPP and Taishan NPP are both 53 km. Daya Bay NPP is 215 km away. Three metropolises, Hong Kong, Shenzhen, and Guangzhou, are also shown.*

The experimental site is at 53 km from the Yangjiang NPP and Taishan NPP. Yangjiang NPP has six reactor cores of 2.9 $GW_{th}$ each (themal power). All cores are the 2nd generation pressurized water reactors CPR1000 [8]. The distances between any two cores of Yangjiang NPP are between 88 m and 736 m. All six cores will be running when JUNO is going to start data taking in 2020. Taishan NPP has planned four cores of 4.59 $GW_{th}$ each. All cores are the 3rd generation pressurized water reactors EPR. The distances between any two cores are between 252 m and 1110 m. The total thermal power of the Yangjiang and Taishan NPPs will be 35.73 $GW_{th}$. It is possible that the last two cores in Taishan will not be available by 2020, in which case the total power will be 26.55 $GW_{th}$ when JUNO will start data taking. Daya Bay complex includes Daya Bay NPP, Ling Ao NPP, and Ling Ao-II NPP in a spread of 1.1 km, each with 2 cores of 2.9 $GW_{th}$. The Daya Bay complex is 215 km away from the JUNO detector, and will contribute about 2.8% of the reactor antineutrino events. There is no other NPP or planned NPP in 500 km around the JUNO experimental site. The thermal power of all cores and the baselines are listed in Table 2.1. The distances from the detector site to the Yangjiang and Taishan cores are surveyed with a Global Positioning System (GPS) to a precision of 1 meter. All these NPPs are constructed and operated by the China General Nuclear Power Group (CGNPG).

| Cores | YJ-C1 | YJ-C2 | YJ-C3 | YJ-C4 | YJ-C5 | YJ-C6 |
|---|---|---|---|---|---|---|
| Power(GW) | 2.9 | 2.9 | 2.9 | 2.9 | 2.9 | 2.9 |
| Baseline(km) | 52.75 | 52.84 | 52.42 | 52.51 | 52.12 | 52.21 |
| Cores | TS-C1 | TS-C2 | TS-C3 | TS-C4 | DYB | HZ |
| Power(GW) | 4.6 | 4.6 | 4.6 | 4.6 | 17.4 | 17.4 |
| Baseline(km) | 52.76 | 52.63 | 52.32 | 52.20 | 215 | 265 |

TABLE 2.1: *Summary of the thermal power and baseline to the JUNO detector for the Yangjiang (YJ) and Taishan (TS) reactor cores, as well as the remote reactors of Daya Bay (DYB) and Huizhou (HZ). [6]*

In absence of high mountains in the allowed area where the sensitivity to the mass hierarchy is optimized, the detector will be deployed in an underground laboratory under the Dashi hill. The activities of the $^{238}$U, $^{232}$Th, and $^{40}$K in the rock around the experimental hall are measured to be 130, 113, and 1062 Bq/kg, respectively. The muon rate and average energy in the JUNO detector are expected to be 0.0030 Hz/m$^2$ and 215 GeV estimated by simulation with the surveyed mountain profile taken into account.

## 2.2 The experimental apparatus

The JUNO detector, shown in Fig. 2.2, consists of a central detector (CD) filled with liquid scintillator (LS) readout by a PhotoMultiplier Tube (PMT) system. VETO detector and a calibration system complete the detector structure. To reach $3\%/\sqrt{E(MeV)}$ energy resolution, the central detector will build a super acrylic sphere and a stainless-steel truss, which will hold 20-kton liquid scintillator, 18000 20-inch PMTs and 25000 3-inch PMTs. The VETO detector will be divided into a top tracker and a water Cherenkov detector. The calibration system will provide different methods for JUNO calibration, with its operation being of a fundamental importance to obtain the required design resolution and to have a checkup of the energy scale of the detector.

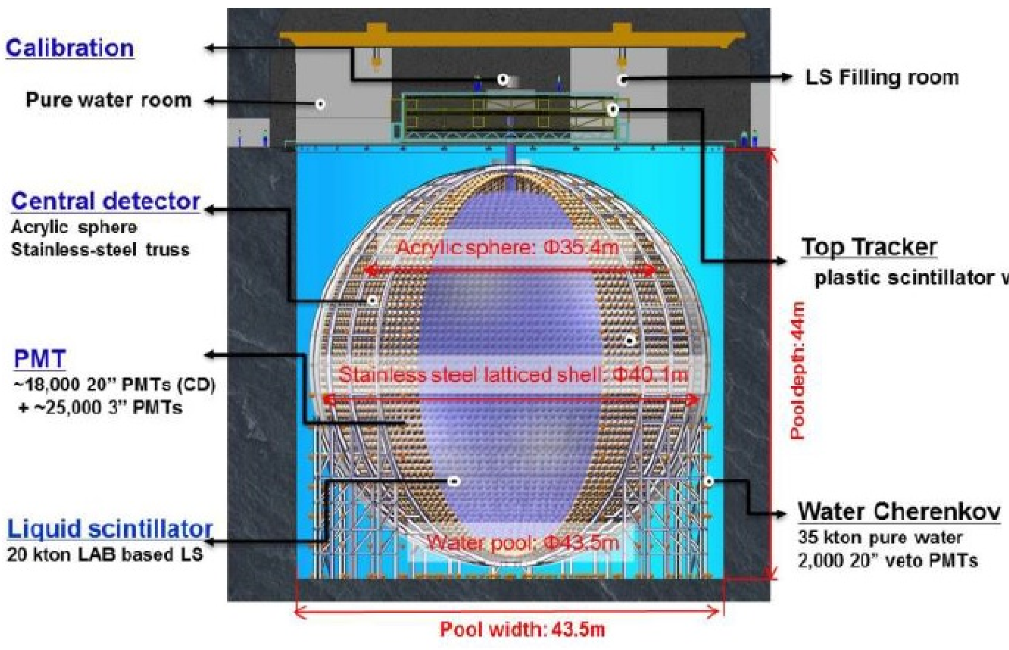


FIGURE 2.2: *The JUNO detector. Main components are indicated.*

### 2.2.1 JUNO detector

To achieve a $3\%/\sqrt{E(MeV)}$ energy resolution is very challenging. To reach the required energy resolution, the following requirements have been set:

- Large PMT photocathode covergage greater than 75
- High PMT photocathode quantum efficiency close to 35%.

- Long liquid scintillator attenuation length greater than 20 m at 430 nm, which corresponds to an absorption length of 60 m with a Rayleigh scattering length of 30 m [28].

The LS has similar recipe as the Daya Bay LS without gadolinium loading. Linear alkylbenzene (LAB), a straight alkyl chain of 10-13 carbons attached to a benzene ring [29], is used as the detection medium due to its excellent transparency, high flash point, low chemical reactivity, and good light yield. The liquid scintillator also consists of 3 g/L 2,5-diphenyloxazole (PPO) as the fluor and 15 mg/L p-bis-(o-methylstyryl)-benzene (bis-MSB) as the wavelength shifter.
The density of the LS is 0.859 g/ml. Twenty thousand ton LS is contained in a spherical container of radius of 17.7 m. The light emitted by the LS is watched by about 18'000 large 20-inch PMTs and 25'000 small 3-inch PMTs. This double set of PMTs is necessary to achieve the resolution requirements of the experiment. In fact, to quantify in a basic and clear approach the resolution characteristics and requirements of JUNO, we can resort to the following simplified relation [7]:

$$\frac{\sigma_E}{E} = \sqrt{\left(\frac{a}{\sqrt{E}}\right)^2 + b^2 + \left(\frac{c}{\sqrt{E}}\right)^2} \tag{2.1}$$

where $a$ represents the stochastic term governed via the overall maximization of light (i.e. through the mentioned coverage, PMT Quantum Efficiency (QE) and features of the LS), and $b$ and $c$ are non stochastic terms controlled by the minimization of systematic effects. Such a control in the detector will be achieved in two ways, by an accurate calibration strategy and by the cross check measurements performed via the auxiliary system of the small 3-inch PMTs. They will allow a full complementarity of the event identification, with particular emphasis on time resolution, spatial reconstruction, dynamic range and triggering strategy. The PMTs are installed on a spherical structure of a radius of 19.5 m, and submerged in a buffer liquid to protect the LS from the radioactivity of the PMT glass.

### 2.2.2 PMT sensors

Photon detectors measure the scintillation light created by interactions of the neutrinos with liquid scintillator and are key components for accomplishing the physics goals of JUNO. Important requirements for photon detectors used in JUNO include high detection efficiency for scintillation photons, large area, low cost, low noise, high gain, stable and reliable, long lifetime.
Two types of large PMTs will be used in the JUNO experiment: the Hamamatsu R12860-HQE and the North Night Vision Technolgies (NNVT) MicroChannelPlate (MCP)-PMT.
The **Hamamatsu R12860-HQE** is a 20-inch large photomultiplier presenting a 460 mm diameter hemispherical bi-alkaline photocathode light-sensitive for wavelength that ranges between 300 nm and 650 nm, with a maximum sensitivity around 400 nm and the relative quantum efficiency around 30% [31]. The chain of amplification has a linear focusing geometry structure, which at 2 kV has a $1 \times 10^7$ nominal gain. The supply voltage foresees a ground cathode and a positive voltage anode. An output pulse is shown in Fig. 2.3
The **NNVT MCP-PMT** comes with a 20-inch glass window and incorporate an MCP in place of the conventional discrete dynodes. The MCP consists in a two-dimensional

array of channels with a 6-20 $\mu$m diameter bundled in parallel. Each channel acts as an independent electron multiplier. The photoelectrons emitted from the photocathode enter the channels of the MCP and impinge on the inner wall where they are multiplied by means of secondary emission. This process is repeated along the channels, and finally a large number of electrons are collected by the anode as an output signal. The quantum efficiency of this photomultiplier is lower than the R12860, 22.86%, while the gain is the same, $1 \times 10^7$.

Figure 2.3 shows a typical large PMT waveform taken from the Hamamatsu and NNVT PMTs. The pulse shape is characterized by a rise time, defined as the time taken by the pulse to change from 10% to 90% of its final value, and by a fall time, defined as the time taken by the pulse to change from 90% to 10% of the pulse amplitude. The temporal parameters are reported in Tab. 2.2 [30]

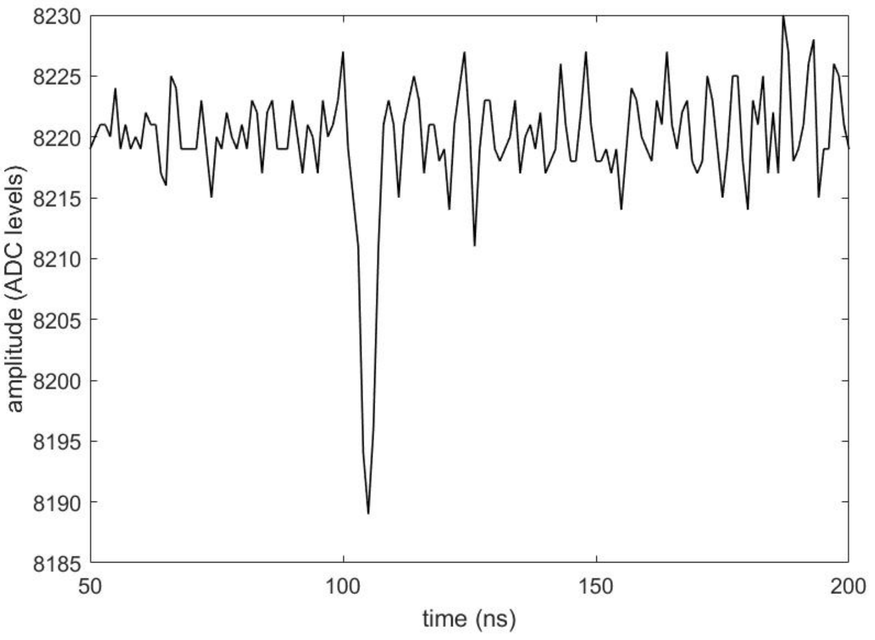


FIGURE 2.3: *Output pulse of Hamamatsu R12860-HQE*

| PMT | $t_r$ | $t_f$ |
|---|---|---|
| Ham. R12860-HQE | $(6.5 \pm 0.5)ns$ | $(13.0 \pm 0.7)ns$ |
| NNVT MCP-PMT | $(2.0 \pm 0.6)ns$ | $(6.5 \pm 0.5)ns$ |

TABLE 2.2: *Temporal parameters of the PMTs.*

Chapter 3

# Readout Electronics

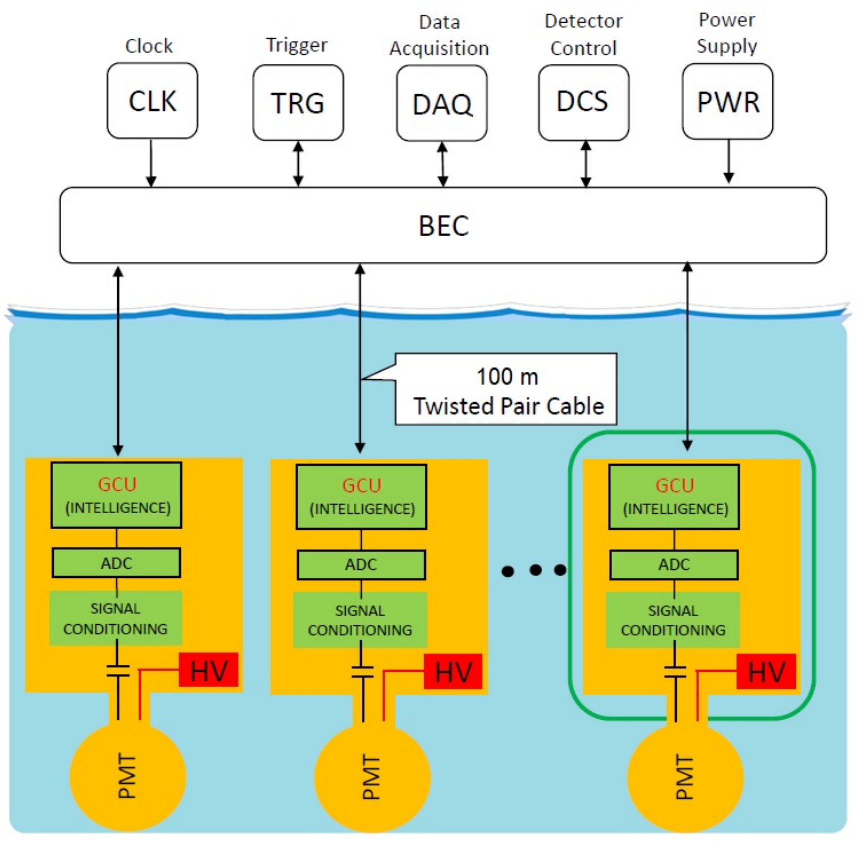


FIGURE 3.1: *JUNO readout electronics architecture. [10]*

The average number of photoelectrons (p.e.) generated by a single large PMT ranges from one p.e. for low-energy events up to thousands p.e. . In both extreme cases, the photoelectrons time profiles have to be determined and the energy of the event has to be measured. The energy represent a fundamental parameter for the neutrino mass hierarchy determination. A basic limitation on the energy resolution arises from the statistics of detected p.e. . This limit must not be worsened significantly by the effect of electronics. For an energy release of 1 MeV in the central detector, which corresponds to an average of 1100 p.e., a 3% design energy resolution is expected.
The layout of the fully-submerged readout scheme is sketched in Fig. 3.1. At the end

of each PMT there is a water-tight housing that contains the essential front-end and readout electronics of the system: the base of the PMT with a module that generates the High Voltage (HV) from a low-voltage input, the Power Board with the responsability of the power distribution and the Global Control Unit (GCU) that, together with the Analogue to Digital Unit (ADU), performs the data digitization, buffering and processing, and, monitors and controls all the relevant parameters. The boards have a circular shape with a 15 cm diameter, as can be seen from Fig. 3.2, in order to fit inside the PMT housing.
The data transfer between the underwater and the backend electronics is achieved through a 100 m long CAT5 cable. Two pairs are reserved for slow control operations and data readout (fast Ethernet), and two are synchronous links for low latency communication between frontend and backend electronics.

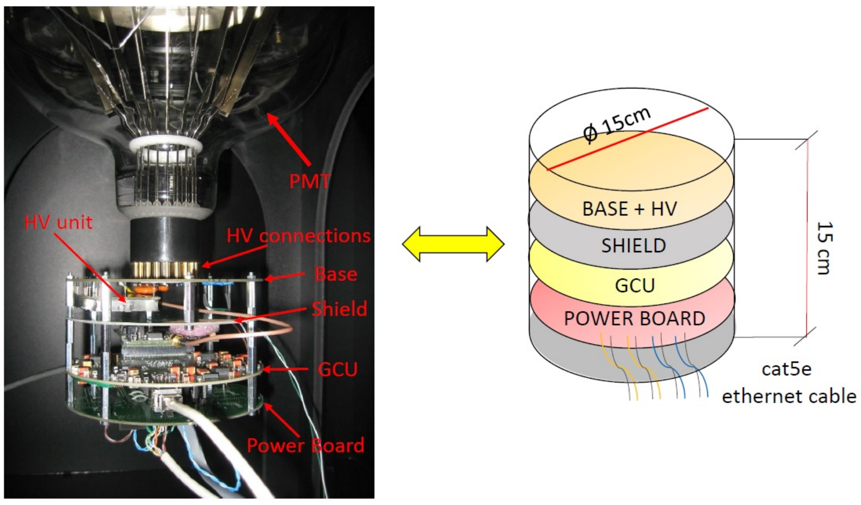


Figure 3.2: *The readout electronics, which forms the Intelligent PMT. The EMI (ElectroMagnetic Interference) shielding protects the GCU from HV unit. [10]*

The readout electronics main task is the reconstruction of the event. Because of the distributed nature of the complete event information, each PMT can only collect one fragment of it. Each of these fragments may be continuously acquired to be further analyzed offline, or they may be pre-analyzed online, inspecting the incoming trigger requests by correlating requests driven by the same event, in order to send a trigger validation signal if it determines that the data is good, acquiring a fixed time window from every single PMT. For a more detailed description about the triggering schemes see Sec. 3.4.
The electronics receives and digitizes the analog signals from the PMTs and transmit all the relevant digital information to the Back End Card (BEC). Each BEC is responsible for the communication between 48 GCUs and the Data Acquisition system (DAQ), the power and the timing and trigger distribution system. To do so, the BEC deals with the fan-out of the signal and guarantees long distance high speed data transfer.
The underwater scheme implements an intelligent PMT concept minimizing the deteriorating effects of analog signal transmission over long cables.

## 3.1 High Voltage Unit and PMT Base

Each PMT is equipped with a voltage divider. The base provides the connections for the High voltage Unit (HV), see (Fig. 3.3) also anchored to the boar.
The high voltage required will be generated directly on the PMT from a 24V voltage input. For each PMT, a single HV potential will be generated, from which all voltages required for the photo cathode, the field shaping electrodes, and any PMT-specific are derived through a voltage divider. The output voltage range goes from 1500V to 3000V. The maximum anode current is 300 $\mu$A. The PMTs will have their cathodes on ground potential, consequently the output of the signal will be on positive HV.
Individual HV units are monitored locally by a micro-controller inside the unit interfaced to the PMT electronic channel, and the parameters are set by the user through the GCU [12].

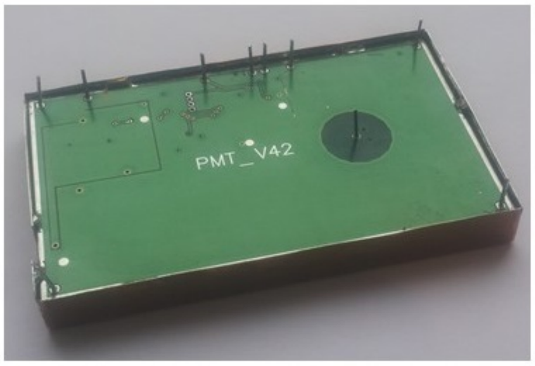


FIGURE 3.3: *High Voltage Unit. The shown pins will connect to the PMT base. [12]*

## 3.2 Global Control Unit

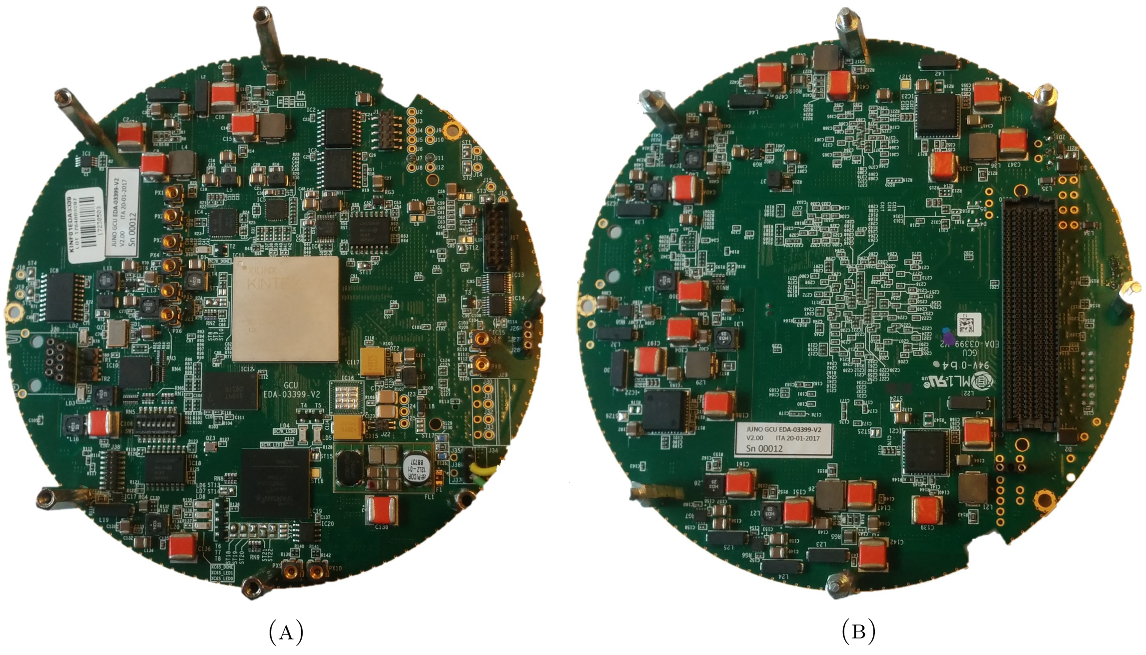


FIGURE 3.4: *GCU top side, panel (A), and bottom side, panel (B).*

The Global Control Unit (GCU), shown in Fig. 3.4, is the intelligent component of the read-out electronics. It's block diagram is shown in Fig. 3.5
It is used to perform the readout of the digitized signals coming from the ADU board and to handle all the data packaging, processing and buffering. It generates the trigger primitives, and stores local events, waiting for a trigger validation signal when running in global trigger mode or immediately sending them up to the BEC when running in auto-trigger mode (the triggering modes will be described in Sec. 3.4). In case of a supernova explosion, the trigger rate for interesting events will increase, probably to a point where the managing of the data readout via Ethernet would be impossible to be sustained; this has led to the decision of implementing a buffer capable to store at least one second of raw data. To perform this task a DDR3 ram memory has been assembled in the GCU.
Both DAQ and slow control operations are carried out via IPbus [16]. The slow controls and monitoring system handle the technical aspects of the experiment, such as high voltages and temperature, reporting and acting on changes in the status of the electronics or its environment, and maintaining the safety of the equipment.
The IPbus is a communication protocol for controlling hardware devices. It consists of a virtual bus with 32-bit word addressing and 32-bit data transfer. The transport protocol used for every user-hardware communication is the User Datagram Protocol (UDP). UDP provides checksums for data integrity, but it has no handshaking dialogues, simplifying the firmware implementation but eliminating any guarantees of delivering and receiving a correct message.
The GCU also provides support for the global synchronization process. All of JUNO's 18'000 GCUs must be synchronized and aligned within a global time in order to correctly timestamp the triggered events. The time accurancy required is 16 ns. The Timing, Trigger and Control system (TTC) [17], developed at CERN, is used as a base for the distribution of synchronous broadcast and individually-addressed messages between the BEC and front-end electronics. The timing protocol under development exploits the TTC system.
The data readout and slow controls use the IPBus protocol via Fast Ethernet.
Since the clock link is exploited also for transmitting messages, the system clock is recovered at GCU level using a Clock Data Recovery (CDR) chip. The board also presents a 62.5 MHz oscillator that provides a local clock signal used for the IPBus transmissions.
The signal processing is carried out by a Xilinx Kintex 7 Field Programmable Gate Array (FPGA), while the reprogramming controller relies on a Xilinx spartan 6 FPGA. For a detailed description on the digital signal processing of the GCU, see Sec. 3.2.2.

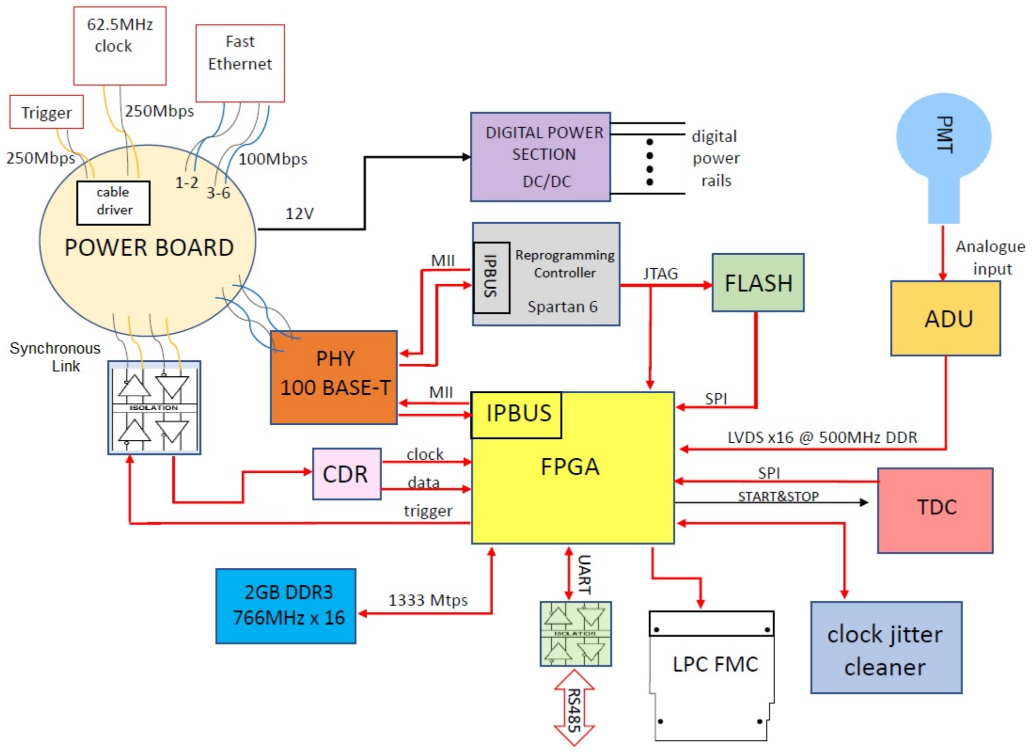


FIGURE 3.5: *Block diagram for the GCU. [10]*

## 3.2.1 Analog to Digital Unit

The Analogue to Digital Unit (ADU) is an Application Specific Integrated Circuit (ASIC) assembled in the GCU, that will digitize the input signal.
The ADU features two TLG121G ADCs, developed by Tsinghua University. A single ADC digitizes at 14 bit with a rate of 1 Gsps. The digital interface is based on a Double Data Rate (DDR) parallel bus. The data is synchronized with a 500 MHz clock. This sampling clock is granted by an external Phase-locked loop (PLL) mounted on the ADU that receives the GCU system clock of 62.5 MHz.
As shown in the schematic of the ADU (Fig. 3.6), the power provided by the GCU consists of three different voltages: 12V, 2.5V and 5.5V; the input analogue signal coming from the PMT is

- provided to the Front End Chip (FEC) component, that has the function of protection for the following electronics and is also a current amplifier ASIC in order to optimize the input voltage range for the ADC.
- the two FEC generated outputs go through a transimpedance amplifier (TIA) to convert the current signal to a voltage signal.
- the voltage outputs are later converted from a single-ended logic to a differential logic using amplifiers.
- the signal is finally digitized by the two ADCs into 14-bits words.
- the 14 bit words coming from each ADC are sent out to the GCU in Low-Voltage Differential Signaling (LVDS) logic.

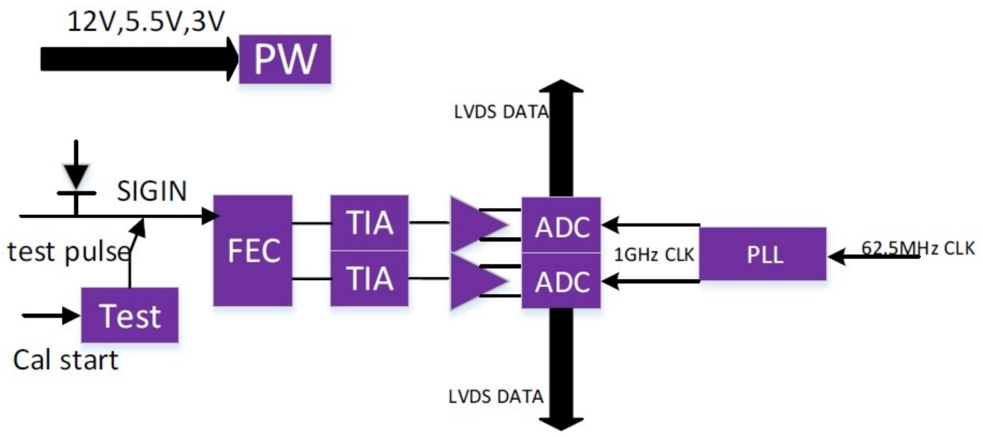


FIGURE 3.6: *Schematics of the ADU Board.*

To assess the performance of the ADU and facilitate its debug, a mezzanine board, shown in Fig. 3.7, were made by the IHEP group in Beijing. This board is connected to the GCU thanks to an FPGA Mezzanine Card (FMC) connector. Through this connection all the signals such as the digitized data, the clock and the power are exchanged between the GCU and to the ADU mezzanine board. The PMT output is directly connected to the ADU via an SMA connector.

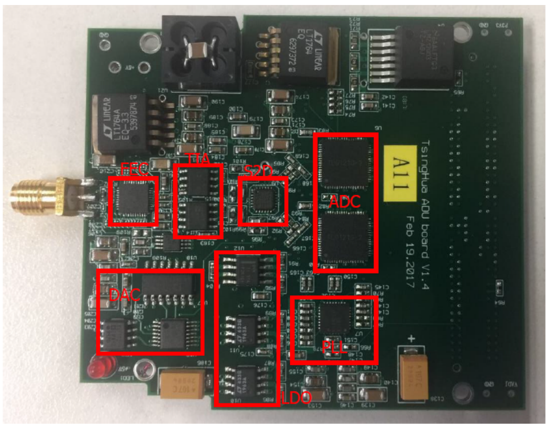


FIGURE 3.7: *Top side of the ADU Board with the main components highlighted. Taken from [13]*

## 3.2.2 Digital signal processing

All of the tasks the GCU board has to execute, are performed on a Field Programmable Gate Array (FPGA) which is an Integrated Circuit (IC) designed to be configured by a customer or a designer after manufacturing (hence field-programmable)[11]. In an FPGA all the operations are computed by its hardware circuitry and components. FPGAs are vastly used in physical facilities and front-end electronics. Some of the advantages using an FPGA are:

- they provide support for fast peripherals control, being a cost-effective solution compared with ASIC designs.

- their technology allows the end user to make changes to their designs very late in the design cycle respect to an ASIC chip. Even after the end of the design phase and production, they can be completely reprogrammed.
- they easily handle real-time tasks.

Reprogrammable silicon also has the same flexibility of software running on a processor-based system, but it is not limited by the number of processing cores available. Unlike processors, FPGAs are truly parallel in nature, so different processing operations do not have to compete for the same resources. Each independent processing task is assigned to a dedicated section of the chip, and can function autonomously without any influence from other logic blocks. To first approximation, the performance of one part of the application is not affected when more processing is added.
The possibility of reconfiguring is due to the modifiable interconnections that allow the user to upload their new design rearranging the elementary logic blocks that compose the FPGA.

For the purposes of the JUNO experiment, a Xilinx Kintex-7 (XC7K160T-2LI FFG676) is used. The chip has been chosen taking into account numerous factors like

- power consumption (4W to 8W depending on the running algorithm)
- cost
- performance
- number of available I/Os
- reliability

The FPGA can be completely reconfigured and controlled through a JTAG-USB connection. Unfortunately it will not be won't be accessible after installation.
To not preclude the reconfiguration possibility and the debugging capability when the board will be underwater, a Xilinx Spartan-6 FPGA is mounted on the GCU. Its firmware has the task to emulate a JTAG connection to the Kintex-7 FPGA over the IPbus protocol. All traffic between the Xilinx's Programming tool (Impact$^{\mathrm{TM}}$) and the parallel port is intercepted by an user space driver that redirects these frames to the virtual JTAG cable. This is represented by the UDP layer, the IPbus core running inside the Spartan-6 FPGA and from here, out on the real JTAG bus passing through some General Purpose Input/Output (GPIO) pins.

## 3.3 Power distribution

The power distribution is relegated to a Power Board (PB) which provides both the GCU and the HV unit with the needed voltages. As shown in Fig. 3.8, the PB receives two 24V, one for analog power and another for digital power. These voltages go through DC/DC converters in order to reach the requirements.
The power board hosts the cable driver and equalizer to support the data transmission over  80m of copper cable.

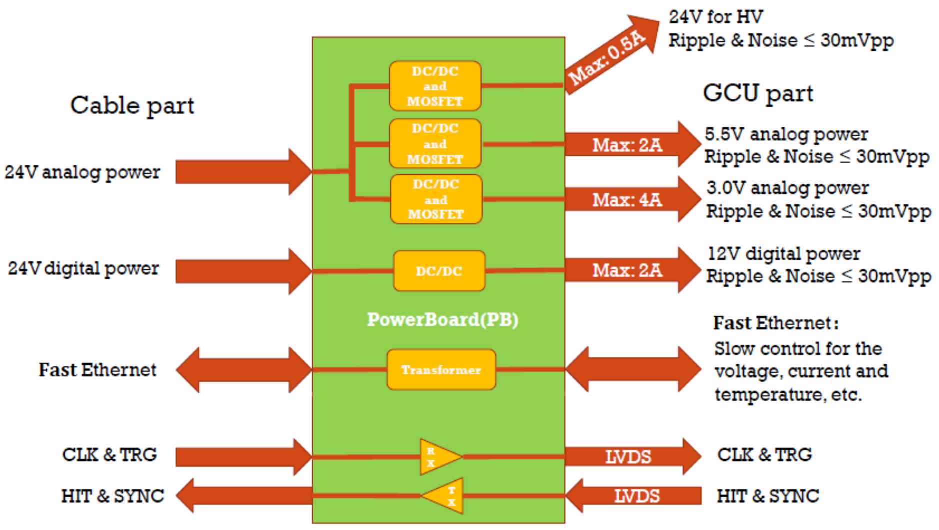


FIGURE 3.8: *Requirements and logical components of the Power Board*

## 3.4 Trigger operation scheme

The JUNO experiment foresees running in two trigger modes: trigger validation mode and auto-trigger mode. In both cases a trigger algorithm is used to identify physical events. Both options are discussed more closely in the following.

In case of **Trigger Validation Mode**, the GCUs generate a trigger request whenever an event is detected above threshold and waits for a trigger validation signal. The information of all PMTs is combined in order to form a global triggering decision that starts the readout of all or part of the PMT signals. The simplest trigger decision is based on multiplicity, therefore information from all of the PMTs above threshold is required. More advanced techniques can be implemented, like the vertex reconstruction system for instance, where informations on PMT location, signal and timing are needed in order to obtain the specific time and place of the interaction of the neutrino with the scintillator liquid. Monte Carlo studies [9] have shown that a simple trigger logic based on the total number of active PMTs within a 300 ns window is sufficient to suppress the background from dark noise random coincidences and to guarantee 100% efficiency for the detection of $\bar{\nu}_e$ events. An important aspect regarding energy reconstruction that potentially favors a global trigger is the fact that this is the only configuration that will allow the recording of waveforms for channels in which the PMT signals are below an individual triggering threshold, i.e. the detection of low-charge photoelectron.

In **Auto Trigger Mode** any event (including dark noise pulses) that exceeds the acquisition threshold for a single photoelectron is buffered for a fixed time window (hundreds of nanoseconds) and transmitted to the BEC.

Chapter 4

# Pulse-Shape Processing for Triggering

In order to improve the detection and trigger efficiency, a method based on pulse-shape analysis [14] has been adapted and designed for our setup. Taking into account the response waveform of the large PMTs used in JUNO, several algorithms have been studied and optimized, with the purpose of implementing them with a modular approach, inside the FPGA.
In these chapter a detailed description of the algorithm is given, explaining how it is possible to extract the energy of a signal peak followed by an exponential decay. However, the purpose of this algorithm must be kept in mind: the aim is not to retrieve a precise value of the pulse energy, but a greater capacity for discriminating small signals, therefore, the evaluation of the algorithm efficiency will not rely on energy measurements, but on discrimination capacity measurements instead.
The pulse-shape analysis is often used for X-ray or germanium detectors signals, to retrieve the energy of the pulse and to reduce the dead time of a measurement. Therefore the signal processing is carried out on impulses characterized by a long exponential decay time and a negligible rise time. In order to be able to exploit the same technique for signals coming from the PMT, we can assume the pulse fall time behaviour as an exponential decay, retrieving its time constant $\tau$ through the formula:

$$\tau = \frac{t_f}{\ln 9} = \frac{t_f}{2.197} \tag{4.1}$$

where $t_f$ is the fall time of the pulse.
Based on the PMTs known fall time [30], the constant decay times are:

$$\tau_{Ham.} = 6\,\text{ns}$$
$$\tau_{NNVT} = 3\,\text{ns}$$

Fig. 4.1 shows the structure of the pulse-shape analysis for triggering which consists of a 4-module pipeline. The first step dynamically tracks the signal baseline (i.e. the average value of the input signal in the absence of pulses) and sets that value to zero to ignore all the fluctuations. It is called baseline follower. The waveform will be further elaborated by the the k$\sigma$ trigger module, that performs a first signal detection in order to stop the baseline from being evaluated when a signal occurs. Afterwards, the data is used as input of the Moving Window Deconvolution (MWD) module, which converts the input pulse into a step function whose amplitude is proportional to the pulse's energy. Proceeding in the chain, the shaping module averages the signal giving it the aspect of a trapezoid or a triangular. This shaped signal is then triggered using a second k$\sigma$ trigger module. In the followinf section MATLAB® examples are shown.

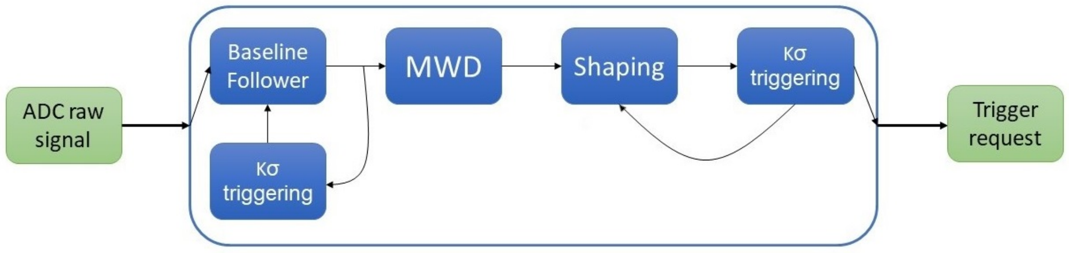


FIGURE 4.1: *Overview of the real-time triggering algorithm*

## 4.1 Signal processing modules

### 4.1.1 Baseline follower and k$\sigma$-triggering

A typical PMT signal is affected by a low frequency amplitude modulation, referred as baseline fluctuation. The module described in this chapter has been developed with the aim of dynamically monitoring the baseline, investigating the mean value in absence of any pulse. To exclude pulses from the baseline computation, a pre-trigger module, the k$\sigma$ trigger, performs a first identification of possible signals.
The dynamical quantities evaluated for every sampling period are:

$$B(t_i) = B(t_{i-1}) + \frac{1}{\tau_b}[R(t_i) - B(t_{i-1})] \tag{4.2}$$

$$\sigma(t_i) = \sigma(t_{i-1}) + \frac{1}{\tau_\sigma}[D(t_i) - \sigma(t_{i-1})] \tag{4.3}$$

where $t_i$ is the discrete sampling time, $R(t_i)$ is the raw signal, $B(t_i)$ the baseline, $\sigma(t_i)$ the standard deviation of the noise and $D(t_i)$ the absolute value of the deviation from the baseline of the raw signal, i.e. $D(t_i) = |R(t_i) - B(t_i)|$. $\tau_b$ and $\tau_\sigma$ are time constants that reflect the dynamics of the evaluated quantities. The value of these parameters must be extrapolated from the actual data stream, as they are dependent on the used PMT, the electronics and the setup. Once extrapolated, the parameters are fixed and can not be further changed.
Equations 4.2 and 4.3 represent Infinite Impulse Response (IIR) digital filters, with a pole set to $p_b = 1 - \frac{1}{\tau_b}$ and $p_\sigma = 1 - \frac{1}{\tau_\sigma}$, respectively.
The main consequence of using IIR filters, is the setting time that these filters need. In fact, to reach a steady state response, they need about 3 to 4 times their setting constants time, that is the corresponding $\tau$ number of samples. This means that, as the baseline follower turns on and starts evaluating, a time of $\max\{4\tau_b, 4\tau_\sigma\}$ has to pass, before the output can be considered stable. This has to be taken into account when triggering on PMT data. Their frequency response and phase response are shown in Fig. 4.2. Fig. 4.32 shows the setting time for both the baseline and the noise evaluation in a VHDL simulation.
The cut-off frequency is dependent on $\tau$, as it gets closer to zero when $\tau$ increases.

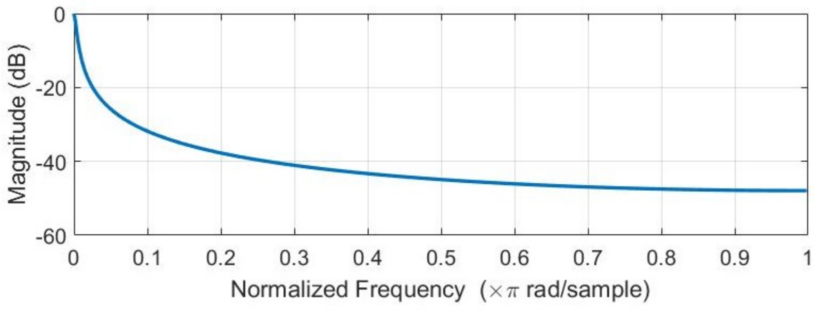


FIGURE 4.2: *Frequency response of the filters used by the baseline follower module obtained using $\tau = 125$. The $\pi$ frequency corresponds to the Nyquist frequency, which is half of the sampling rate of a discrete signal processing system.*

To correctly evaluate the baseline of the raw signal stream, any pulse must be recognised as it occurs, in order to freeze the baseline value and resuming its dynamic evaluation when the pulse returns to the zero value. This job is carried out by the k$\sigma$ trigger module. This module works as a simple threshold trigger,constantly evaluating the signal $D(t_i)$. The trigger level $T = k\sigma$ is dynamically checked during the experiment and the user can set the parameter $k$ to express the threshold level in units of $\sigma(t_i)$, the standard noise deviation. In Fig. 4.3 we can see how the k$\sigma$ threshold works. The gray line in the top part of the plot, represents the raw data, consisting of pulses on a slowly fluctuating baseline with random noise superimposed. The black line represents the deviation of the raw signal from its baseline in terms of standard deviation from the noise.
As long as the signal remains above the level, the baseline follower stays in "sleeping mode", waking up only when the signal goes below the threshold.

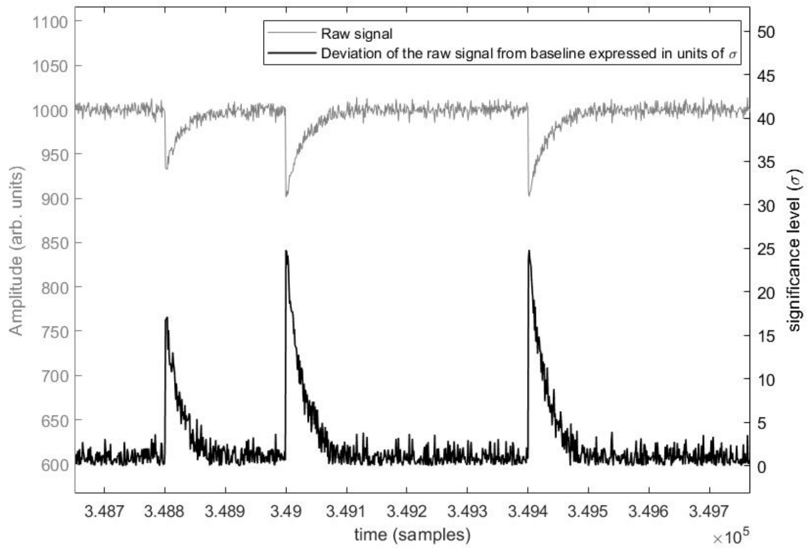


FIGURE 4.3: *Schematic representation of the k$\sigma$ triggering. The black bottom line, with values presented on the right vertical axis, shows the deviation from the baseline in units of $\sigma$; The gray top line is the digitized signal, with values presented on the left vertical axis.*

### 4.1.2 Moving window deconvolution

One of the main advantages of the adopted trigger scheme is to be able to discriminate noise spikes from real signals, even in the case where both of them are beyond the same threshold. As shown in Fig. 2.3, the detector signal is characterized by a fast rising edge followed by an exponential decay. The Moving Window Deconvolution algorithm (MWD) [23] is used to exploit these characteristic to improve the detection efficiency. A detailed discussion of the MWD algorithm can be found in [24].
From an exponential decay signal, starting at the time $t_0 = 0$, we can model its amplitude at the time $t$ by the expression

$$A(t) = \begin{cases} N\exp(-\frac{t}{\tau}) & t \geq 0 \\ 0 & t < 0 \end{cases} \tag{4.4}$$

where $\tau$ is the decay constant and $N$ the maximum amplitude. We can define a new function $U(t_k)$, with $t_k > 0$, expressed in terms of the initial amplitude $N$ as

$$\begin{aligned} U(t_k) &= A(t_k) + N - A(t_k) \\ &= A(t_k) + N\left(1 - \exp\left(\frac{-t_k}{\tau}\right)\right) \\ &= A(t_k) + \frac{1}{\tau}\int_0^{t_k} A(t)dt \\ &= A(t_k) + \frac{1}{\tau}\int_{-\infty}^{t_k} A(t)dt \end{aligned} \tag{4.5}$$

where the last step expanding the integral from 0 to $-\infty$ is permitted because the amplitude $A(t_k)$ is zero for $t_k < 0$. $U(t_k)$ aims to obtain an Heavyside step function multiplied by the initial amplitude of the pulse $N$.
Eq. 4.5 is only valid for positive times. We can easily extend it for negative times giving it the form

$$U(t) = \begin{cases} N & t \geq 0 \\ 0 & t < 0 \end{cases} \tag{4.6}$$

With the goal of implementing this algorithm inside an FPGA, we can transform the expression in Eq. 4.5 to the digital domain, retrieving the value $N$ at time $t_k$

$$U(t_k) = A(t_k) + \frac{1}{\tau}\sum_{i=-\infty}^{k-1} A(t_i) \tag{4.7}$$

where $\tau$ is now expressed in units of the sampling time.
Adapting this equation to our signal we obtain:

$$U(t_k) = x(t_k) + \frac{1}{\tau}\sum_{i=-\infty}^{k-1} x(t_i) \tag{4.8}$$

where $x(t_k) = B(t_k) - R(t_k)$.

The MWD filter is derived from differentiating the expression in Eq. 4.8

$$\begin{aligned} MWD(t_k) &= U(t_k) - U(t_{k-w}) \\ &= x(t_k) - x(t_{k-w}) + \frac{1}{\tau} \sum_{i=k-w}^{k-1} x(t_i) \end{aligned} \tag{4.9}$$

where $w$ is the width of the moving window. The aim is to make the positive level of the step function last for only the number of samples expressed by $w$
As can be seen from Fig. 4.4, the application of the MWD technique to an exponential decay will result in a step function, with a flat top as long as the width of the window $w$. For PMTs signals, the exponential decay is considerably fast, making the rise time of the signal not negligible. Thus, the rising edge of the transformed signal will not be sharp, but it will present a leading edge that reflects the shape of the rising edge from the original digitized signal. The almost-step function will anyway reach a plateau value, that is the same for a given deposited energy inside the detector, independent from the rise time.
Fig. 4.5 shows the effects of a wrong estimation of $\tau$. If $\tau$ is underestimated, the step function will not present a flat top, but a positive slope instead (Fig. 4.5b). This will lead to an overestimation of its amplitude and, as a consequence, of its energy, which is obtained in the shaping module by averaging the flat top. Moreover, the step function will present a tail. On the other hand, using an overestimated $\tau$, will lead to an underestimation of the amplitude of the step because of a negative slope at its top (Fig. 4.5c). The waveform would also present an overshoot below the zero value, reaching it after long time.
The width of the window is also a very important parameter. In fact, if the rising time of the signal is greater than the window $w$, the step function will never reach the flat top, inducing an erroneous estimation of the energy. Using a value too large, however it will increase the dead time for the measurement, since as long as the step function is high, the energy reading from an occurring pulse is not reliable.

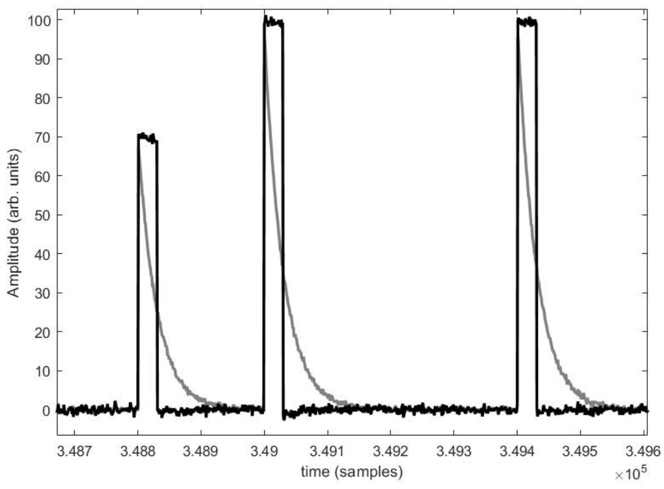
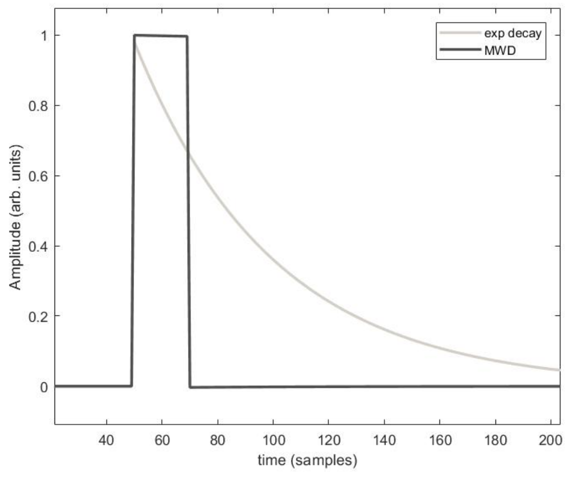
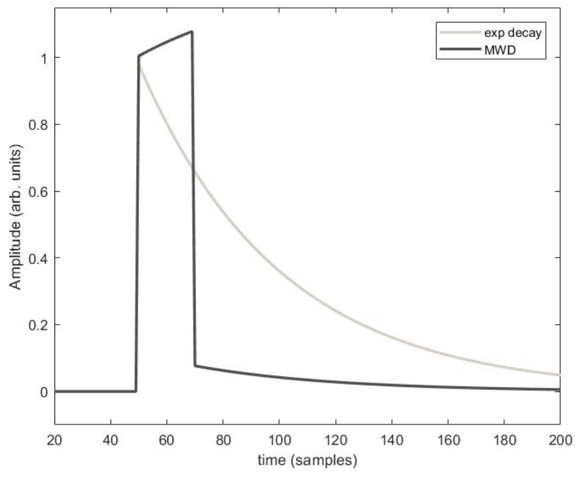


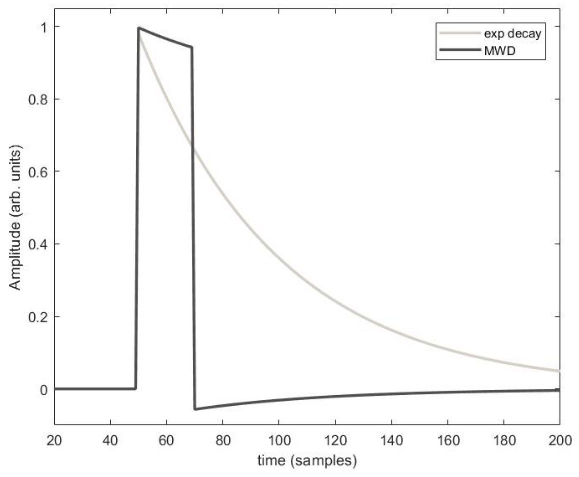
FIGURE 4.4: *Monving Window Deconvolution applied to simulated pulsed with a decay time constant $\tau = 30$ and a window length $w = 30$. The pulses have noise superimposed.*



(A) $\tau$ = *real decay time*



(B) $\tau$ *underestimated by 20%*



(C) $\tau$ *overestimated by 20%*

FIGURE 4.5: *Moving Window Deconvolution. The gray line corresponds to the original exponential decay, whereas the black line corresponds to the result of the MWD algorithm. A 20% mis-estimation of* $\tau$ *leads to a mis-estimation of the energy of about 3%.*

### 4.1.3 Shaping

The amplitude reached by the step function at the output of the deconvolution module is a good estimate of the pulse energy, even if a non-negligible rise time, the approximate exponential decay and a non negligible noise component make the flat top of the step function not well defined, as shown in Fig. 4.9. Therefore, the goal of the shaping module is to average the plateau value, in order to obtain a unique and well-defined value. This shaping is carried out by a simple Moving Window Average filter (MWA). The MWA is the most common filter in Digital Signal Processing, mainly because it is the easiest digital filter to implement. In spite of its simplicity, the moving average filter is optimal for a common task: reducing random noise while retaining a sharp step response. As the name suggests, the filter operates by averaging a fixed number of points from the input signal to produce one point in the output signal. This procedure is expressed by the following equation:

$$MWA(t_k) = \frac{1}{l}\sum_{j=1}^{l} MWD(t_{k-j}) \tag{4.10}$$

where $x(t_k)$, $x(t_{k-1}) \dots x(t_{k-l})$ are the signal at the sample times $t_k$, $t_{k-1}...t_{k-l}$, $MWA(t_k)$ is the output signal at time $t_k$, and $l$ is the width of the filter window, equal to the number of points in the average.

For an easier FPGA implementation, Eq. 4.10 can be rewritten as

$$MWA(t_k) = MWA(t_{k-1}) + \frac{1}{l}(MWD(t_k) - MWD(t_{k-l-1})) \tag{4.11}$$

Through this module, the user has the possibility to select between two different shapes. If $l$ is less than $w$ the output will be shaped as a trapeze, with a $w-l$ long flap top. Instead, if $l = w$, the shape will be the one of a triangle (Fig. 4.6). The advantage of using a trapezoidal form is a greater discriminatory power because of its characteristic plateau; however, if a triangular shape is chosen, the module filtering efficiency will be maximum.

Once the module has shaped the deconvolved output, a second noise evaluation (Eq. 4.3) on the newly formed signal will be performed, followed by a k$\sigma$ trigger that provides the final trigger of the algorithm. The threshold for this final trigger is independent from the one used by the k$\sigma$ trigger in the baseline follower module, to allow different discrimination capacities.

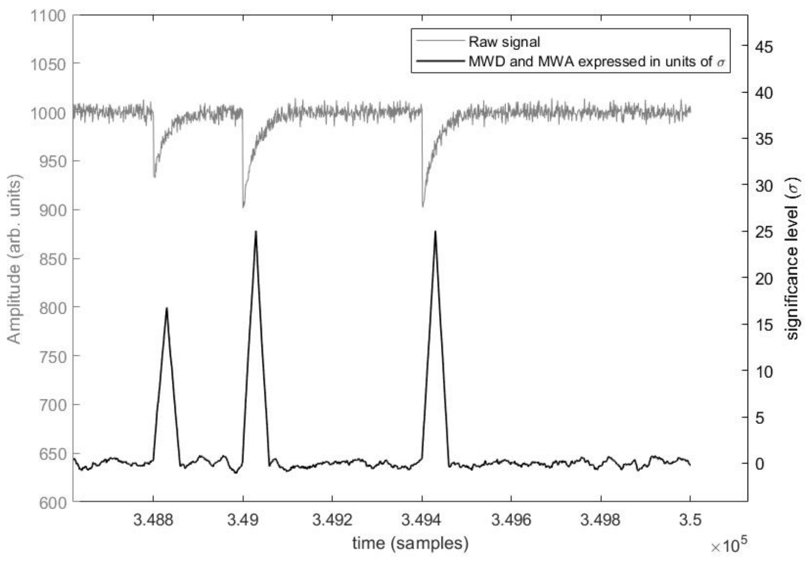


Figure 4.6: *Moving Window Average, black bottom line, applied to the deconvolved output of Fig. 4.4 with the right vertical axes showing the deviation in units of $\sigma$. The gray top line shows the simulated pulses with the right vertical axes showing the amplitude values. The window length for the average filtering is set at $l = 30$, setting the triangle shape*

### 4.1.4 MATLAB® simulation

The algorithm has first been tested in the MATLAB® environment. To run the program a real PMT output has been registered and uploaded into a MATLAB® matrix. The amplitude of the pulse has been changed and adapted to match several possibilities that may happen during the actual experiment.
First, a characterization of the pulse had to be performed, extracting the decay time constant. The parameter must be known and fixed to tune the algorithm accordingly. The PMT pulse used in the simulation is showed in Fig. 4.7

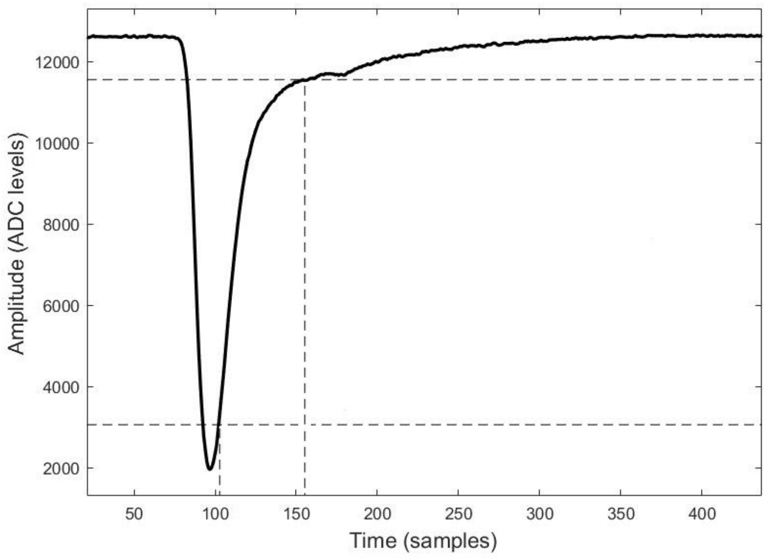


FIGURE 4.7: *PMT pulse used for the MATLAB® simulation. The horizontal dashed line correspond to the 90% and the 10% of the pulse amplitude. The vertical line corresponds to the start and the stop of the fall time $t_f$.*

The fall time, $t_f$, and the relative decay time constant, $\tau$, are:

$$t_f = 56 \text{ samples} \implies \tau \simeq 26 \text{ samples} \tag{4.12}$$

From the analogue data flow, the parameters $\tau_b$ and $\tau_\sigma$ of the baseline follower module must be tuned. A small value guarantees a fast response with a negligible setting time, but the filtering power will be minimized having a wider window for the permitted frequencies. A larger value, however, ensures a greater filtering power with a lower cut-off frequency, but the response time increases, as well as the setting time. When choosing the parameters, $\tau_b$ and $\tau_\sigma$ must match the dynamics of the baseline and of the electronics background noise fluctuations respectively, in order to allow the algorithm to follow any variation of the stream of data. The values chosen for the simulation are the following:

| | |
|---|---|
| $\tau$ | 26 samples |
| $\tau_b$ | 100 samples |
| $\tau_{sigma}$ | 500 samples |

TABLE 4.1: *Temporal parameters used for the MATLAB® simulation.*

But for a real implementation into the FPGA, these parameters will be mapped in the configuration space, accessible to the end user. The final set up is dependent on the PMT characteristic baseline and noise variations. Fig. 4.8 shows the output of the baseline follower (black line) on the raw data (gray line).

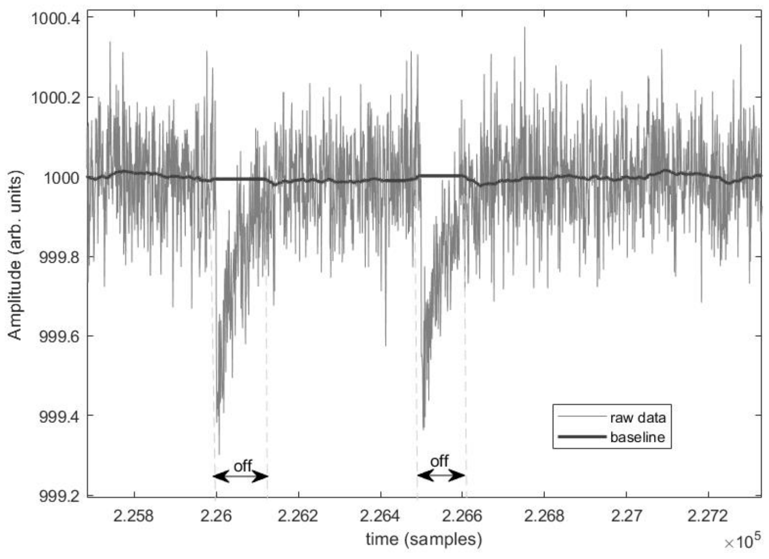


FIGURE 4.8: *The output of the baseline follower $B(t_i)$ (black line) on the synthesized raw data (gray line). The timing windows where the follower is "off" are indicated.*

Fig. 4.9 shows the application of the MWD to the original sample. The figure underlines how a direct PMT pulse is far from being an ideal exponential decay. The significant rise time makes the deconvolved signal's step go well beyond its amplitude. Moreover, the non-horizontal plateau and the long tail are all signs of deviations from the fitting curve (There are no values for the decay time constant $\tau$ that would fix it). However, it will not affect the algorithm, as its job is only to trigger, and not to precisely measure the energy of the pulse. Nonetheless, these imperfections can be shaped away thanks to the following Moving Average filter, whose output is a triangle. This can be seen in Fig. 4.10. The MWD and MWA windows length are both set to 50 samples.

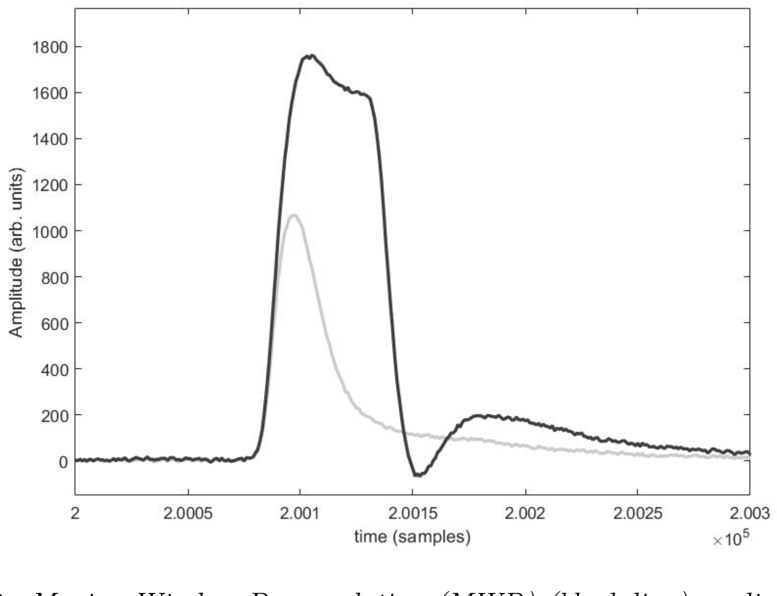


FIGURE 4.9: *Moving Window Deconvolution (MWD) (black line) applied to the baseline $B(t_i)$ minus the raw signal of a real pulse (grey line)*

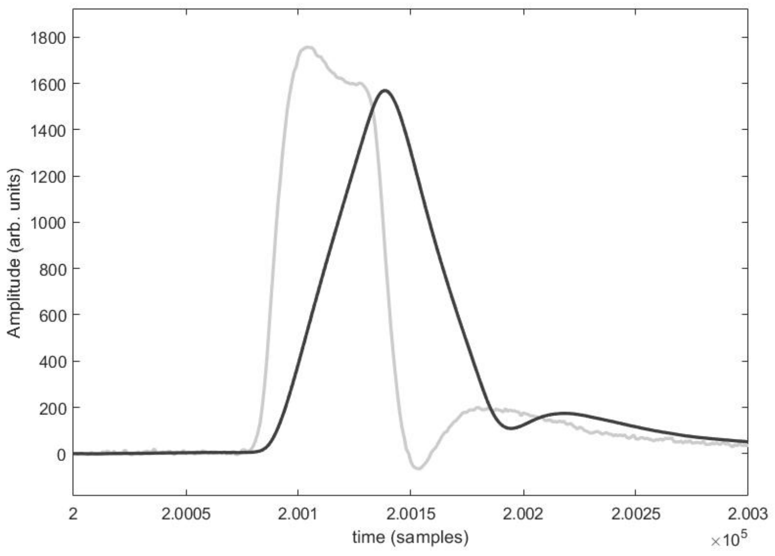
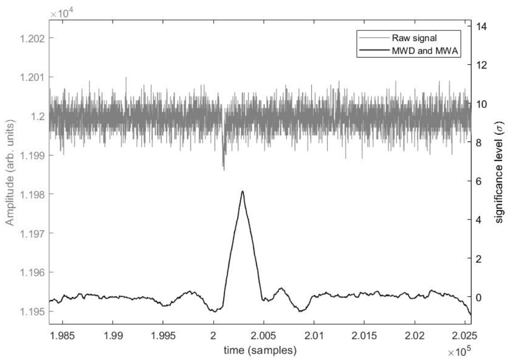
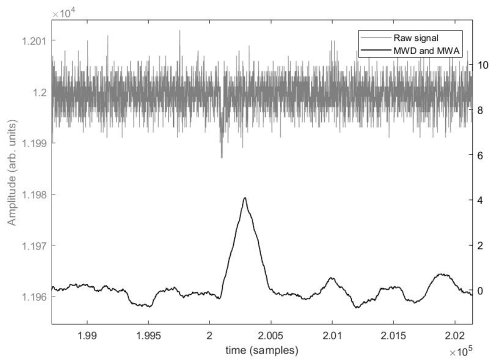


FIGURE 4.10: *Moving Window Average (MWA) (black line) applied to the deconvolved pulse (gray line) of Fig. 4.9*

Simulating the algorithm in MATLAB® using different amplitudes for the pulses, points out the advantages of using this algorithm rather than a simple leading-edge threshold trigger. As shown in Fig. 4.11, the algorithm enhances the presence of a pulse, allowing a clear triggering even though noise spikes reach almost the same amplitude as the pulses.



(A)



(B)

FIGURE 4.11: *MATLAB® simulations using different amplitudes for the input pulse. Notice how, discriminating the triangle (bottom black plots, right vertical axes) at $4\sigma$, there are no false triggers; however, with a simple leading-edge threshold for the raw signal (top grey plots, left vertical axes), trigger requests would be sent even in absence of pulses.*

## 4.2 Trigger digital implementation

The goal of this work is to design a system capable of receiving the raw data stream from the PMT detector and generate a trigger signal that goes high as soon as a pulse is recognized.
The project has been developed using the Xilinx ISE Project Navigator v14.7 [18], running under Ubuntu 16.04. The programming language used for the whole algorithm is the VHSIC Hardware Description Language (VHDL), where VHSIC stands for Very High Speed Integrated Circuits. VHDL allows the user to create and model combinatorial and synchronous logic blocks. An hardware description language like VHDL, is a textual description consisting of expressions, statements and control structures. For this reason, programming in VHDL has some similarities with programming software languages, such as C; however, these similarities stop at the visual level, as the results of these two programming present many substantial differences. Several are:

- VHDL language describes the architecture and behavior of discrete electronic systems. It describes synchronous and asynchronous circuits and opens the possibility to create hierarchical designs. All the statements are concurrent unlike sequential computing languages such as BASIC, C, python. Sequential statements, in VHDL, may appear only in processes.
- Synchronous hardware circuits are synchronized by a clock signal. This makes the timing analysis being of crucial importance while programming.
- The VHDL describes and models complex logical systems and opens the possibility to perform behavioral simulations before synthesis tools translate the design into real hardware (gates and wires)

There is a variety of USB-JTAG Xilinx programming solutions on the market. These programming solutions allow the final user to download the design (bitstream) directly inside the FPGA and/or to store the configuration file inside an Serial Peripheral Interface (SPI) or Byte Peripheral Interface (BPI) flash memory: at boot time, the FPGA readouts the default configuration file from the EEPROM (Electrically Erasable Programmable Read-Only Memory) attached.

### 4.2.1 Digitized data flow

As described in Sec. 3.2.1, the raw data is sent to the FPGA with a 14 Gbps bit rate. Each ADU outputs a 14-bit data bus in Double Data Rate (DDR) format, synchronized with a 500 MHz clock. The raw digitized stream presents a baseline at a positive value that allows data to be rapresented by the unsigned data type despite the negative nature of the signal pulses. Both the data and the clock signals enter the FPGA in Low Voltage Differential Signaling (LVDS) logic. LVDS was chosen because of its capability to consume very little power compared to other signaling technologies. Since closing the timing of a synchronous circuit clocked with a 500 MHz double data rate clock, would be very difficult, the first stage encountered inside the firmware for each data bit, is a source-synchronous deserializer core. The deserialization function is achieved by a Serializer/Deserializer (SerDes) block, the ISERDESE2 [19] core readily available in the 7 series FPGA family. This tile consists in a serial-to-parallel converter with specific clocking and logic features designed to facilitate the implementation of high-speed source-synchronous applications.

The behaviour of the ISERDESE2 core is customizable by the end user via a set of hard-coded attributes. Through this choice, has been defined that

- The incoming data stream will be processed at DDR; in fact the data will be sampled by the ISERDESE2 with a 500 MHz clock both on its rising and falling edge.
- The width of the serial-to-parallel converter is fixed at 8. This means that 8 consecutive 14-bit samples are parallelized in a 128-bit vector. This makes the ISERDESE2 an 8:1 deserializer, utilizing the Q1:Q8 outputs (Fig. 4.12).

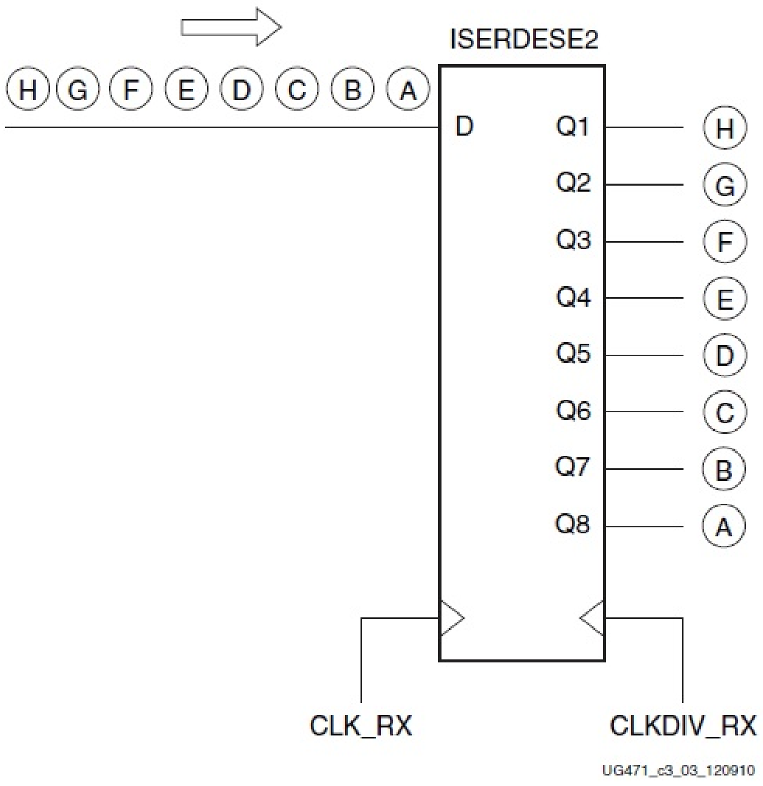


FIGURE 4.12: *Bit ordering on Q1-Q8 outputs of ISERDESE2 ports. [19]*

The serial **Input Data** for the ISERDESE2, after being converted into single ended signals, goes through a delay block. To avoid marginal capturing and to correctly sample the 14-bit data bus coming from the ADU, delay elements have been used. The Series 7 FPGAs have the possibility to use the IDELAYE2 [19], which is a 31-taps, 78 picoseconds each, delay primitive with a calibrated tap resolution that allows incoming signals to be delayed on an individual input pin basis. In fact, the data and the 500 MHz clock may arrive to the FPGA with an unknown phase relationship, that may cause metastability whenever the setup and hold times are violated. Moreover, the delay elements can be used to correct propagation delay mismatch that potentially lead to unwanted spikes on the sampled waveform. (Fig. 4.13). During testing, the control on the IDELAYE2 has been performed through the software Chipscope Pro using the Virtual Input/Output (VIO) module.

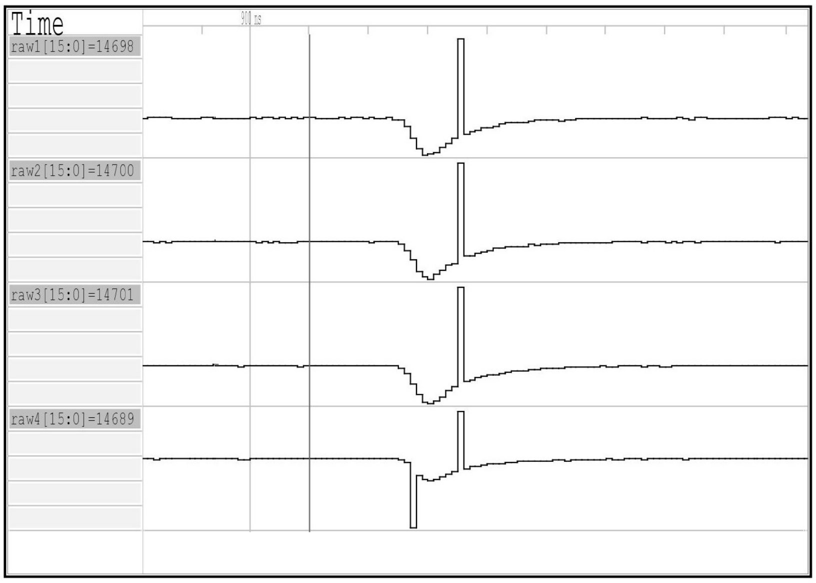


FIGURE 4.13: *Spikes caused by an incorrect sampling in the FPGA. The waveforms are obtained through the software Chipscope Pro and visualized in the software GTKWave.*

The **Input clock** ports for the ISERDESE2 consists of two different clock signals: one clock signal is dedicated to clocking of the high frequency input serial data stream, while the other clock signal drives the slow output of the serial-to-parallel converter. After the parallelization performed by the ISERDESE2, the bits on the 128 bit bus are reordered by the **Bit ordering module**, placing the eight 16 bits wide samples in a row, as shown in Fig. 4.14.
The module also takes care of of the bit swapping: during the routing of the board, several bits have been swapped in order to minimize the number of VIAs (Vertical Interconnect Access, an electrical connection between layers in a physical electronic circuit that goes through the plane of one or more adjacent layers). The module simply invert the value of the bits involved.

The whole digital data sampling block diagram is shown in Fig. 4.15.

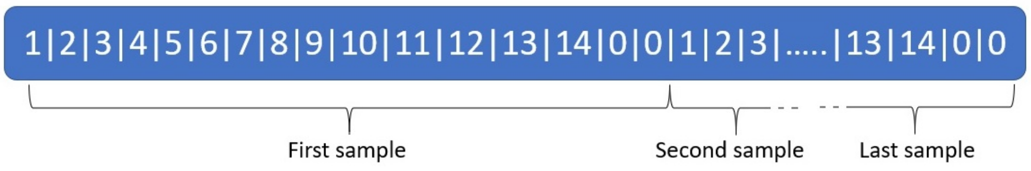


FIGURE 4.14: *Scheme of the bit adjustment by the ordering module.*

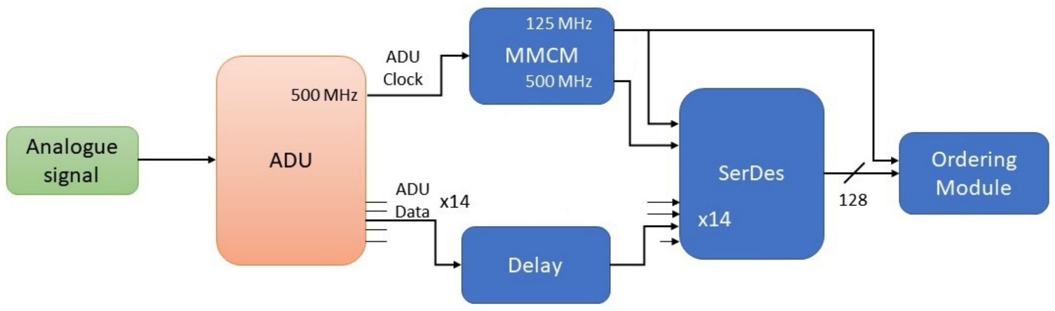


FIGURE 4.15: *Data sampling from the ADU inside the FPGA*

### 4.2.2 Real number approximation

An important change that occurred when implementing the project into the FPGA from the MATLAB® algorithm concerns the data type. MATLAB® software, in fact, processes the data through floating-point arithmetics. For an hardware implementation floating point math is typically expensive in terms of FPGA resources and coding work. Therefore, an implementation based on fixed-point arithmetics has been taken into consideration.
When converting a floating-point implementation to fixed-point,the optimal fixed-point data types must be identified, meeting the constraints of embedded hardware while satisfying system requirements for numerical accuracy. To achieve this, The MATLAB® tool Fixed-point Designer™ [25] has been used.
First of all the code needs to be prepared for an FPGA implementation, optimizing a few aspect that will make the algorithm more reliable and easy-to-code. The signal processing performed by the algorithm includes multiplications and divisions of the internal signals by a coefficient, for example when executing the MWD (Eq. 4.9) and MWA (Eq. 4.10) algorithms. This can be carried out using multipliers or dividers provided by the CORE generator. To achieve a lower-latency trigger response, the coefficient can be "tweaked", changing them to the nearest power of 2, allowing the division to be made only by shifting the bit vector. The error introduced by this approximation is acceptable because the aim of this trigger module is to increase the trigger efficiency and not to accurately measure the pulse energy.
After preparing the code for the fixed-point conversion, a simulation is performed with the Fixed-point Designer™ tool, allowing us to estimate the error introduced by using the fixed point arithmetic. The input pulse used as input raw data is shown in Fig. 4.16:

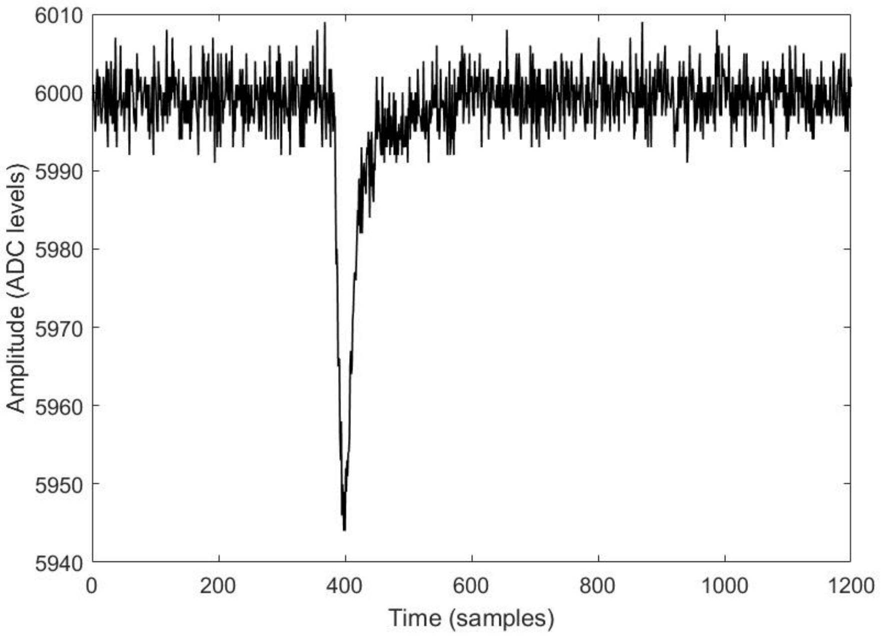
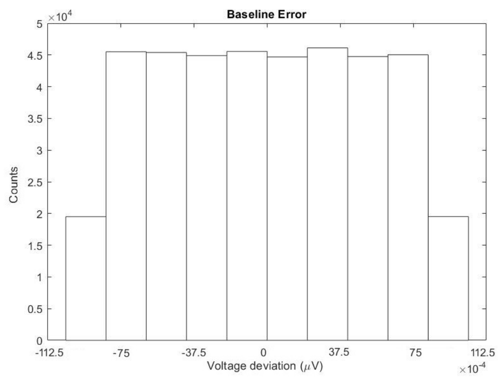
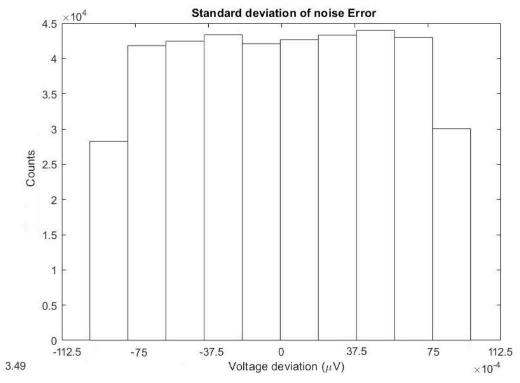


FIGURE 4.16: *Pulse analyzed by the Fixed-Point Designer$^{TM}$ tool.*

Through this software, we have the possibility to create an histogram, visualizing the error introduced on the different steps of the triggering algorithm. This error consists in the deviation of the plots created in floating-point data type with the ones created in fixed-point data type. For the fixed-point math the optimal word length has been found to be 32 bit wide with 12 bits for the fractional part. The error introduced by the data-type conversion in the baseline follower is showed in Fig. 4.17.



(A)



(B)

FIGURE 4.17: *Histogram showing the deviation for every sample from the floating-point data type to the fixed-point data type for the baseline plot (a) and the sigma plot (b) (Sec. 4.1.1). The plots have been obtained using 32 bit fixed-point implementation with 12 bits reserved for the fractional part.*

As we can see from these plots, the maximum error introduced using the fixed-point implementation for the baseline follower is 112.5 mV, which is negligible compared with the quantization noise introduced by the ADC.

The same verification has been performed on the MWD and on the MWA part of the algorithm. Histogram of Fig. 4.18 shows the maximux deviation of the triangular plot obtained using fixed-point math from the one obtained using floating-point math.

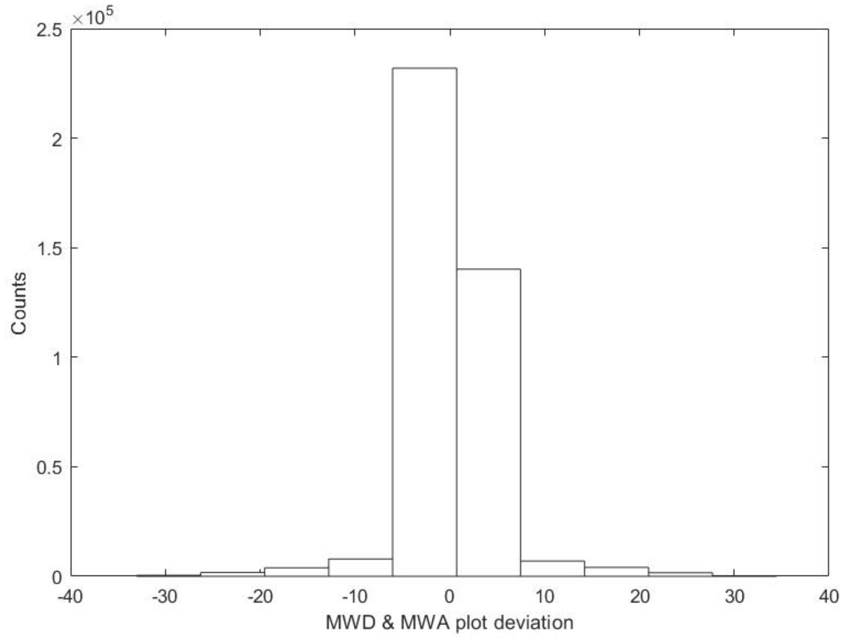


FIGURE 4.18: *Histogram showing the deviation for every sample from the floating-point data type to the fixed-point data type for the MWD and MWA plot. The plots have been obtained using 32 bit fixed-point implementation with 12 bits reserved for the fractional part.*

In order to check if these values may have an influence to our implementation, a conversion in energy units is needed. However, since the input pulses used for the MATLAB® simulations have identical rise time and fall time, the energy is proportional to their voltage amplitude value, which can be our alternately comparison parameter to declare this "data-type changing error" negligible.
In order to convert the triangular peak values to a pulse amplitude voltage value, a few measures of the two quantities are obtained, changing the input pulse amplitude. The results are plotted in Fig. 4.19:

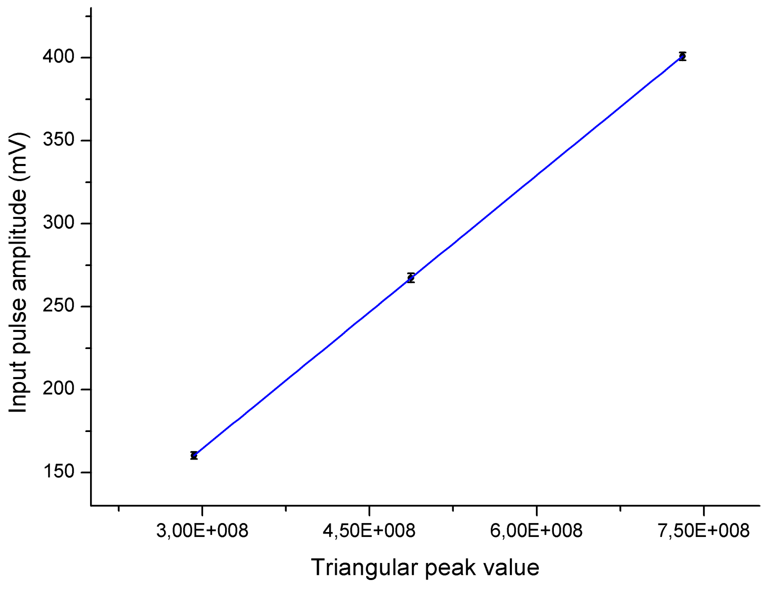


FIGURE 4.19: *Relationship of the MATLAB® triangular peak value to the input pulse amplitude. A linear fit, blue line, is performed.*

The results of the linear fit are:

| $Pulse\ amplitude = a + b \cdot (Triang.\ peak\ value)$ | |
|---|---|
| $a =$ | $(0 \pm 3)\, mV$ |
| $b =$ | $(5.49 \times 10^{-7} \pm 7 \times 10^{-9})\, mV$ |

FIGURE 4.20: *Linear fit parameters.*

Based on these results, a difference in the MWD-MWA plot of 30, corresponds to a pulse amplitude difference of $(1.65 \times 10^{-5} {\pm} 2 \times 10^{-7})$ mV, which is negligible compared to the quantization noise introduced by the ADC.

### 4.2.3 Clock domain synchronization

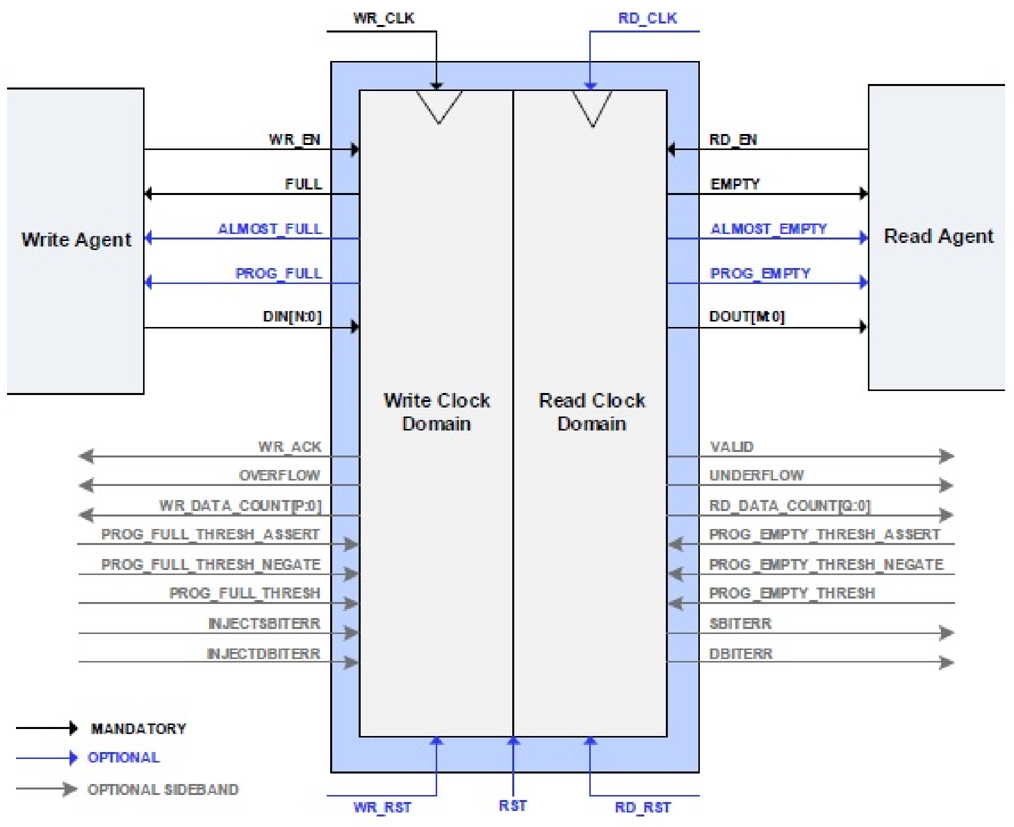


FIGURE 4.21: *Native interface FIFOs signal diagram. Any optional or optional side-band port are not used in the implemented FIFO, except for the RD_CLK and RST ports. [22]*

The cross clock domain synchronization problems have been addressed using dual port FIFO memories generated using the ISE Core Generator wizard.
The mandatory attribute for the FIFO to achieve the clock domain synchronization is the independent reading and writing clock; the independent clock configuration of the FIFO Generator allows to implement unique clock domains on the write and read ports, with no requirements on the phase and frequency relationship. The signal diagram for the I/O ports is shown in Fig. 4.21.
Regarding the FIFO dimensions, the write and read width was selected to be of 128 bits in order to store an entire 8:1 parallelized raw signal, while the depth of the FIFO is 33 words. The depth is not really an important parameter for this implementation, as data are continuously written and readout with the same frequencies of 125 MHz, therefore the FIFO is never full. As additional features an asynchronous reset has been selected.

### 4.2.4 Low-pass polyphase decimator filter

As explained in Sec. 4.2.1, the FPGA can not sustain a working clock of 1 GHz that would be needed to process every single sample, as closing the timing would be very difficult. For a first implementation of the trigger algorithm, the analysis of one

sample out of four would be a good compromise between clock frequency (250 MHz) and resolution.
To reduce the sampling rate from 1 Gsps to 250 Msps, a decimation of a factor 4 is needed. This is achieved through a decimation filter [15]. Decimation by an integer factor can be explained as a 2-step process:

- Reduce the high-frequency signal components with a digital low-pass filter to avoid aliasing.
- Downsample the filtered signal by $M$; that is, keep only one every M$^{\text{th}}$ sample.

These steps are shown in Fig. 4.22, where the low-pass filter has the impulse response $h(n)$ that z-transforms to $H(z)$. It needs to be implemented with a Finite Impulse Response (FIR) design not to deform the signal, thanks to its linear phase response.

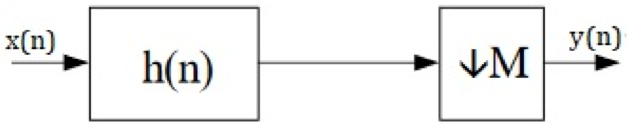


FIGURE 4.22: *Low-pass and decimator filter.*

With this straight-forward filter, the output obtained with a filter length of 95 coefficients will be:

$$y(n) = \sum_{k=0}^{95} h(k) \cdot x(n-k) \tag{4.13}$$

This filtering is not efficient, as it also computes the output values that the decimator filter will reject. To improve the efficiency we can make the filter estimate only one sample out of $M$, which are the outputs that will not be eliminated by the decimator. To clarify further, let's suppose that $L = 2$ and that the filter $h(n)$ has a length of $N = 4$ coefficients. we presume that the decimator filter will reject all of the outputs with an odd index. The output will be (index will be in subscript to facilitate the reading):

$$y_{2i} = x_{2i}h_0 + x_{2i-1}h_1 + x_{2i-2}h_2 + x_{2i-3}h_3 \tag{4.14}$$

We can see how the even input samples are convoluted with a "subfilter" characterized by the coefficients with an even index, while the odd input samples are convoluted with a "subfilter" characterized by the coefficients with an odd index:

$$\begin{aligned} y_{2i} &= \{x_{2i}h_0 + x_{2i-2}h_2\} + \{x_{2i-1}h_1 + x_{2i-3}h_3\} \\ &= \sum_{k=0}^{1} x_{2i-2k} \cdot h_{2k} + \sum_{k=0}^{1} x_{i-(2k+1)} \cdot h_{2k+1} \end{aligned} \tag{4.15}$$

Now, knowing that $\mathcal{Z}(h(2k)) = H(z^2)$ and $\mathcal{Z}(h(2k+1)) = z^{-1}H(z^2)$, the Eq. 4.15 can be visualized in Fig. 4.23.

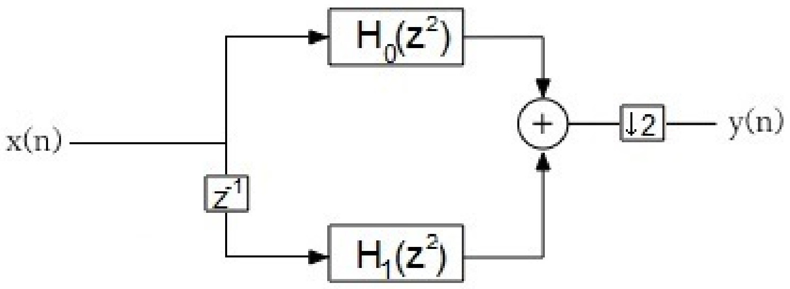


FIGURE 4.23: *Decimator filter that refers to the Eq 4.15. $z^{-1}$ equals to a delay of one sample.*

By transporting the decimation to the left of the filters, the real polyphase decimator filter is obtained, as shown in Fig. 4.24.

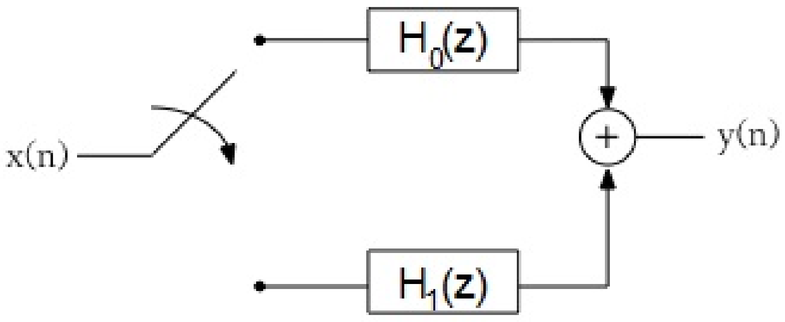


FIGURE 4.24: *Polyphase decimator filter with a decimation of a factor $M = 2$; The decimation is shown as a simple switch. The two filters, $H_0(z)$ and $H_1(z)$ operates at a frequency that is half the sampling rate of the input.*

In general, undersampling with a ratio of 1:$M$ let us splits the input in $M$ subsequences, each of which is convoluted with a different $L/M$ long "subfilter" working ot a frequency that is $1/M$ the sampling rate of the input, obtaining a polyphase filter like the one in Fig. 4.25

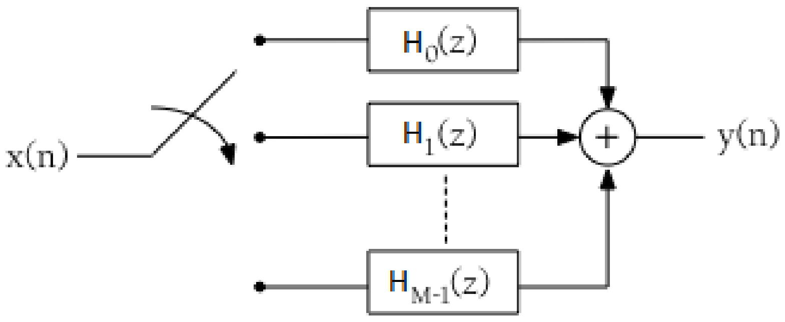


FIGURE 4.25: *Polyphase decimator filter with a decimation of a factor $M$.*

The filter has been implemented in the design using the Xilinx® LogiCORE ™ IP FIR Compiler, which provides an interface for the user to generate FIR filters. The polyphase design has a decimation factor $M = 4$, meaning that from the 8-samples word we have as the input, we want two single words as the output. To achieve this, two identical filters need to be working in parallel, each one processing a 4-samples word. Every filter is composed by four subfilters working with different coefficients. The block diagram for the polyphase decimator filter is shown in Fig. 4.26.

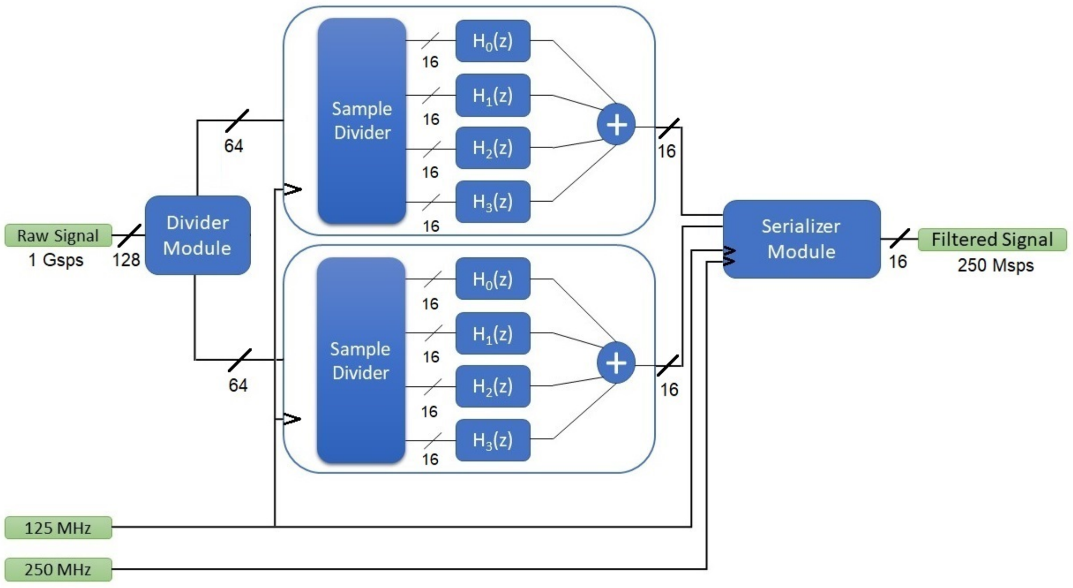
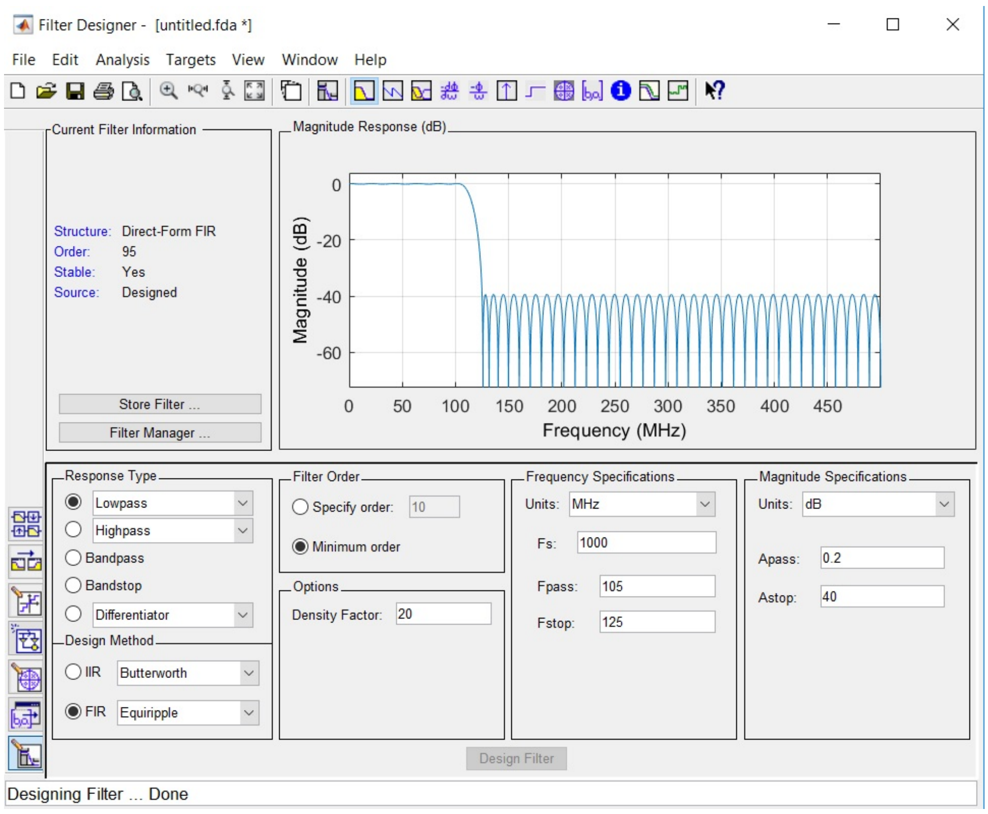


FIGURE 4.26: *Block diagram of the polyphase decimator filter. The sample divider in our design acts like the switch in Fig. 4.25.*

The FIR Compiler allows the designer to upload the needed coefficients during the CORE generation, through a .coe file. These coefficients can be divided into a fixed number of subsequencies, selecting the working one when instantiating. In our design, four subsequencies are needed. The generation of the file together with the coefficients is relegated to the MATLAB® tool filterDesigner [26]. This tool allows the user to choose any wanted specification such as the type of response and frequency requirements. The design and specifications of the filter is shown in Fig. 4.27.



Current Filter Information

Structure: Direct-Form FIR
Order: 95
Stable: Yes
Source: Designed

Response Type: [✓ Lowpass] Highpass Bandpass Bandstop Differentiator

Design Method: IIR Butterworth [✓ FIR Equiripple]

Filter Order: Specify order: 10 [✓ Minimum order]

Options: Density Factor: 20

Frequency Specifications: Units: MHz; Fs: 1000; Fpass: 105; Fstop: 125

Magnitude Specifications: Units: dB; Apass: 0.2; Astop: 40

Designing Filter ... Done

FIGURE 4.27: *filterDesigner window with the chosen specifications.*

The order of the filter is 95. Consequently before reaching a steady-state response, 95 samples have to be processed.
After the generation of the coe file, the coefficients must be converted in fixed-point words of 16 bits and reordered for the FIR compiler to split them correctly. The effect of the floating-point - fixed-point conversion is shown in the filterDesigner window, adding a dedicated plot in the magnitude response. It is negligible.

The output consists in two 16 bits words at a frequency of 125 MHz. The algorithm needs to process these words in series, at 250 MHz. A module has been designed in order to serialize the two samples.

### 4.2.5 Double baseline follower

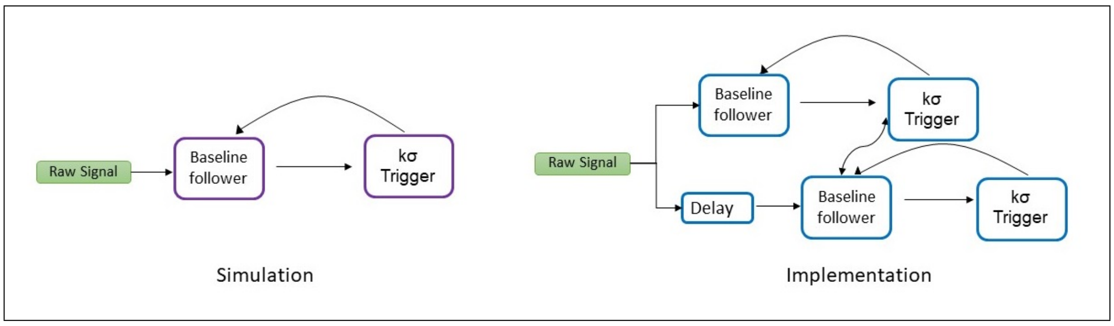
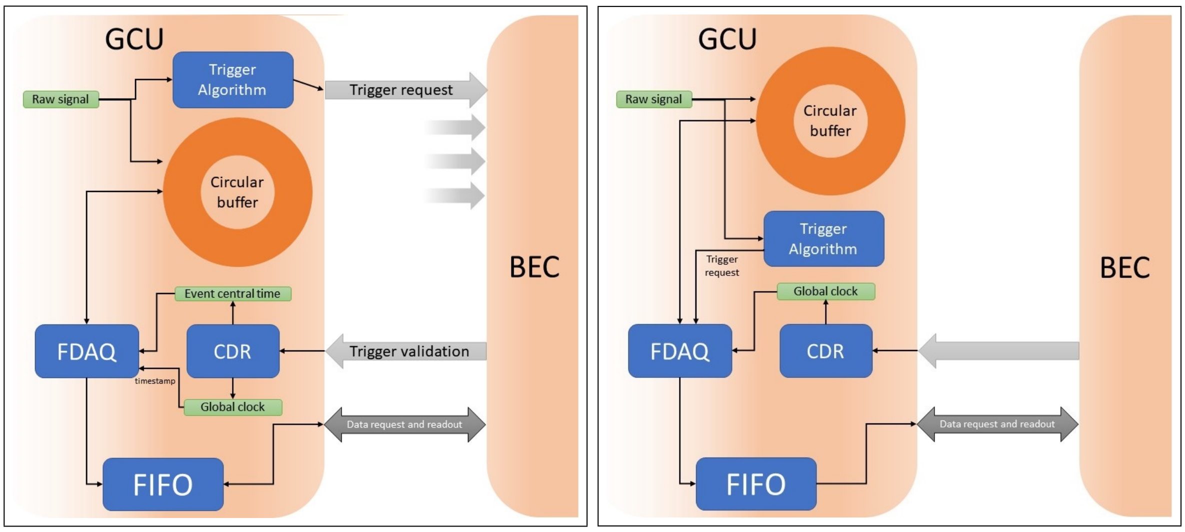


FIGURE 4.28: *Differences between the MATLAB® and the FPGA implementation for the baseline follower and for the kσ trigger modules.*

Every operation in an FPGA is synchronized by a clock signal, consequently the algorithm presents delay between the evaluation of the baseline and the check of the k$\sigma$ trigger. Thus, if a pulses is present in the raw digitized signal, the k$\sigma$ trigger module can freeze the baseline evaluation (Eq. 4.2) from the baseline follower only after this delay time has passes, letting the pulse influence the baseline value. To avoid this, a double baseline follower and k$\sigma$ trigger has been designed, as shown in Fig. 4.28.
The first baseline follower and k$\sigma$ trigger work exactly like the MATLAB® simulation (Sec. 4.1.1); The second baseline follower module has the same raw signal as input, but delayed of a number of samples at least equal to the computational delay introduced between the first baseline estimation and the first trigger check. The second baseline follower may have its output frozen by the first k$\sigma$ trigger (which exploits only the first baseline evaluation, i.e. without any delay on the raw signal) and by a second module of k$\sigma$ trigger (which exploits the delayed baseline evaluation). With this design, the second baseline will not be affected by any baseline pulling by a pulse, as the first k$\sigma$ trigger will suspend the evaluation before the signal will actually arrive, thanks to the delay introduced, and the second will hold the suspension for the whole duration of the signal.

The same design has been used for the evaluation of the standard deviation of the high frequency noise (Eq. 4.3) from the raw signal.

### 4.2.6 Data readout



(A) *Data readout in trigger validation mode.* (B) *Data readout in auto trigger mode.*

FIGURE 4.29: *Block diagrams for trigger validation mode and auto trigger mode. The light grey arrows stand for communications through the synchronous links, while the darker bottom arrow stands for an IPBus communication.*

The only GCU connection with the out-of-water electronics is through the CAT5E cable, where two twisted pairs are used as a fast Ethernet and the remaining two as synchronous links.

The data readout in **Trigger validation mode** is showed in Fig. 4.29a. The raw signal is both continuously registered on a circular buffer, which is able to contain tens of $\mu$-seconds, and processed by the trigger algorithm. Each BEC receives trigger

requests from 48 GCUs. These are forwarded to the trigger validation unit, which processes them and returns a trigger validation message. This message is broadcasted to all GCUs and contains the validated event central time. This information, alongside the global system clock recovered at GCU level by a CDR, is exploited by the FPGA Data Acquisition System (FDAQ) to extract the corresponding data window from the circular buffer.
After adding an header containing all information needed for the off-line data elaboration, the event is delivered to an IPBus slave FIFO memory readable by software.

When working in **Auto-trigger mode** (Fig. 4.29b), there is no need for any trigger validation message from the BEC. In this case the raw data is still stored inside the circular buffer, but it is readout by the FDAQ as soon as the trigger algorithm generates a trigger request. Even in auto-trigger mode data is readout via IPBus. The worst case scenario, in terms of data readout throughput, is determined by dark noise, and, it is well in the range of fast Ethernet.

The size of the event read in the circular buffer by the FDAQ, together with other parameters, is configurable via slow control through IPBus, and the values are stored in the configuration registries.

The need for a low latency communication channel, the synchronous links, along side other reasons, is caused by the limited size of the internal buffer of the FPGA. In fact, as mentioned before, its capacity is limited to tens of $\mu$-seconds of raw data story, contrary to the fast Ethernet latency which is around tens of m-seconds.

## 4.3 Simulation results

In this section the results of the algorithm simulations are shown. The simulations are performed using the Xilinx software ISE Simulator (ISim). The tests are performed using a real PMT output (Fig. 4.30) registered in a text file, varying the pulse amplitude with MATLAB® . The VHDL test bench code is attached in the Appendix.

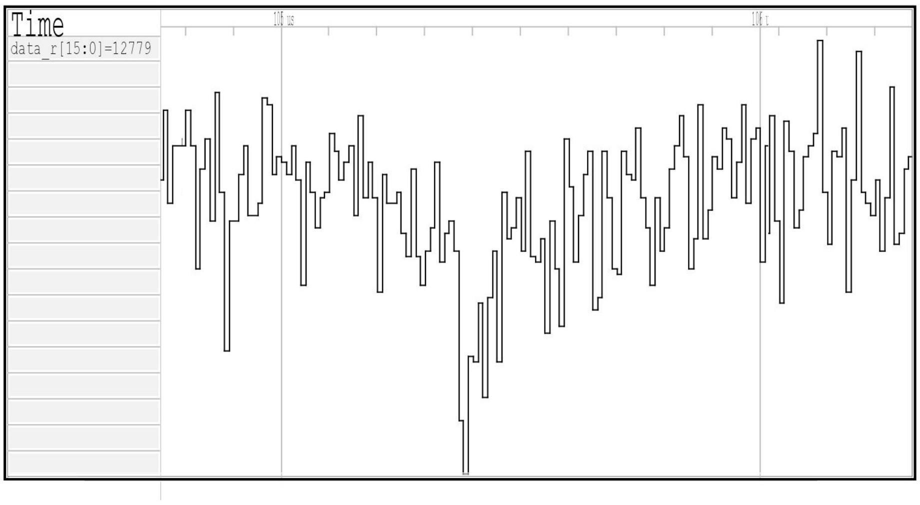
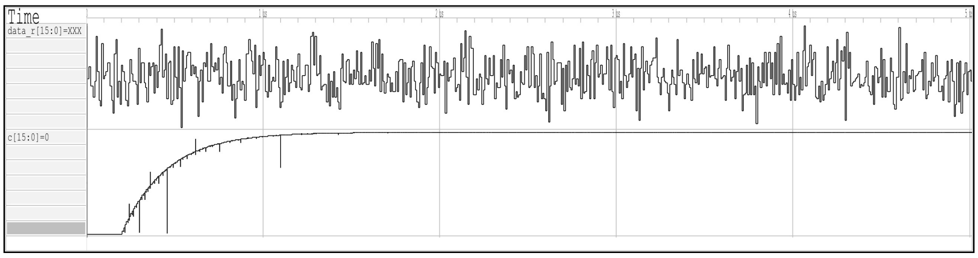


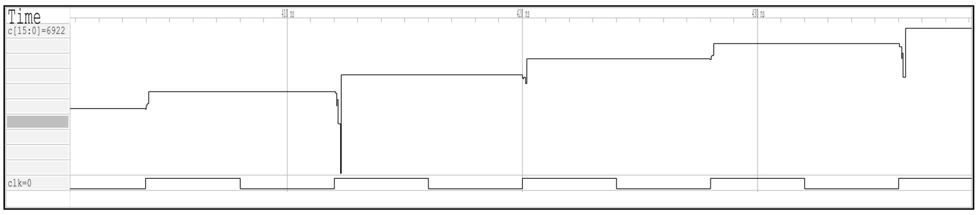
FIGURE 4.30: *Pulse-shape of the signal used as input for the test-bench. Different amplitudes and different noise magnitudes have been used during testing.*

In this section we only analyze the trigger algorithm presented in the thesis, the integration with the firmware already written for the GCU will follow.
The graphics shown here refers to behavioural simulations of the algorithm, as the post-route simulations, although presenting identical results at the rising edge of the synchronizing clock, introduce short spikes, compared to the clock period, caused by the different processing time of the combinational logic for the different bits that make up the signal. Although not being a problem itself, as the data is valid at the rising edge of the clock signal, the reading of the graphs is muddled, as can be seen from Fig. 4.31.
In the simulation of the algorithm the clock domain synchronization module (Sec. 4.2.3) and the low-pass polyphase decimator filter (Sec. 4.2.4) have been tested separately, as the creation of a test-bench for those modules exploiting a real PMT signal would have introduced further difficulty (caused by the parallelized nature of data by the SerDes, Sec. 4.2.1) without adding a relevant utility; in fact their performance is not related to the waveform of the input signal.

In Fig. 4.32, 4.33, 4.34 and 4.35, different simulations of the algorithm are shown. The waveform are visualized in the software GTKWave [27].



(A)



(B)

FIGURE 4.31: *Spikes on the Post-Place & Route simulation. As shown in figure B, the spikes are present for a brief time, far less than the whole clock period, needed for the bits to adjust to new value. In this example, the baseline evaluation, "c", is reported. Input data, "data_r", is not affected by this problem because defined in the test-bench file.*

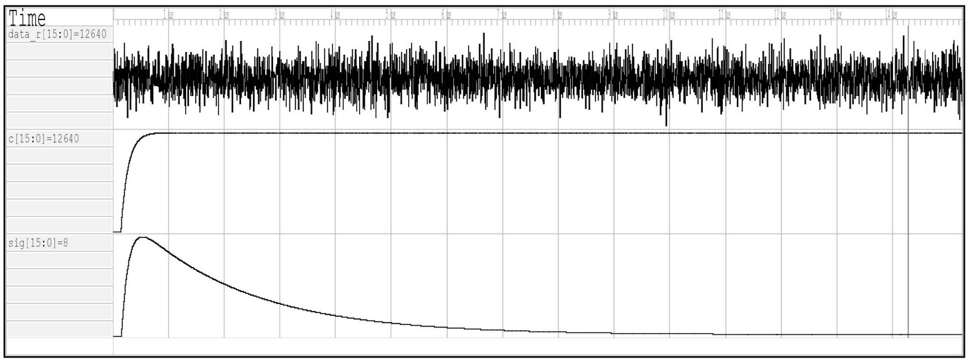


FIGURE 4.32: *Setting time for the baseline evaluation, signal "c", and for the standard deviation of the noise, signal "sig", referred to the input file "data_r". As shown, the setting time for the baseline is about 1 $\mu$-second, while for the noise evaluation is about 10 $\mu$-seconds. These times can be set by changing the time constants of the baseline follower; in this example $\tau_b = 32$ and $\tau_\sigma = 512$.*

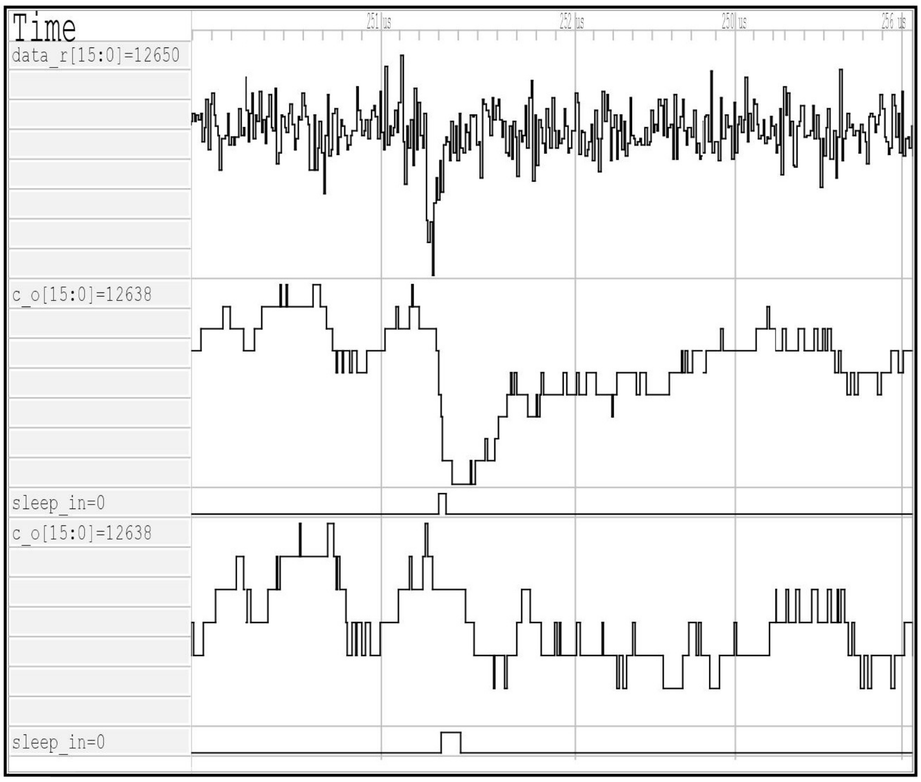


FIGURE 4.33: *Results of the double baseline follower implementation and action of the $k\sigma$ trigger. The "data_r" represents the raw input signal, the top "c_o" and "sleep_in" represents the baseline evaluation from the first baseline follower and the freezing signal from the first $k\sigma$ trigger respectively. The bottom "c_o" and "sleep_in" represent the counterpart from the second baseline follower and $k\sigma$ trigger. As shown, the second baseline follower is immune to the baseline pulling from the pulse.*

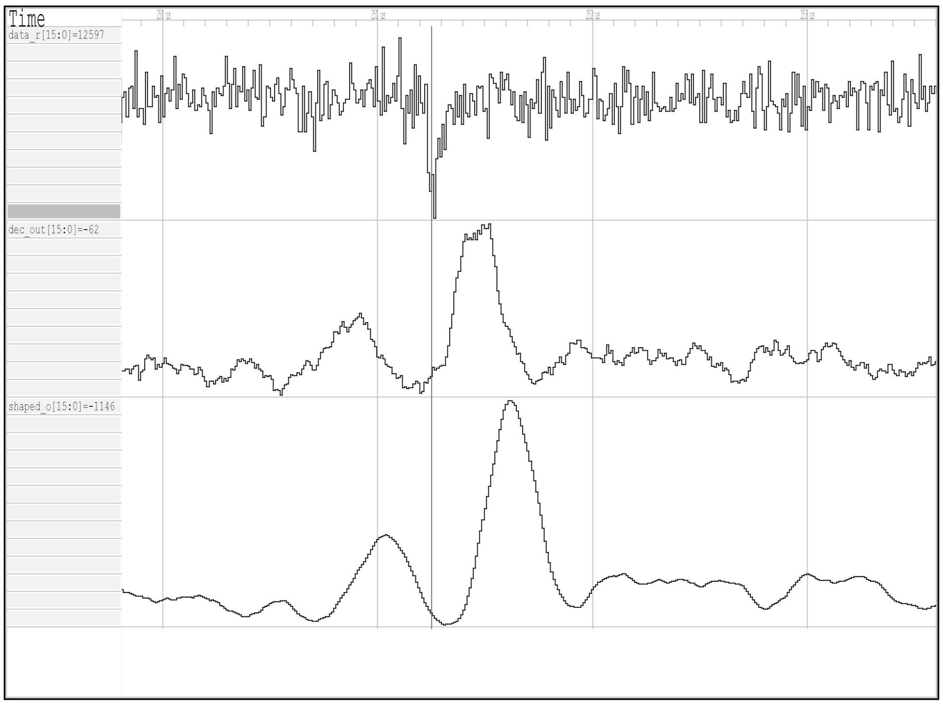


FIGURE 4.34: *Results of the moving window deconvolution (MWD), "dec_out", and the triangular shaping thanks to the moving window average (MWA), "shaped_o", operated on the input signal "data_r".*

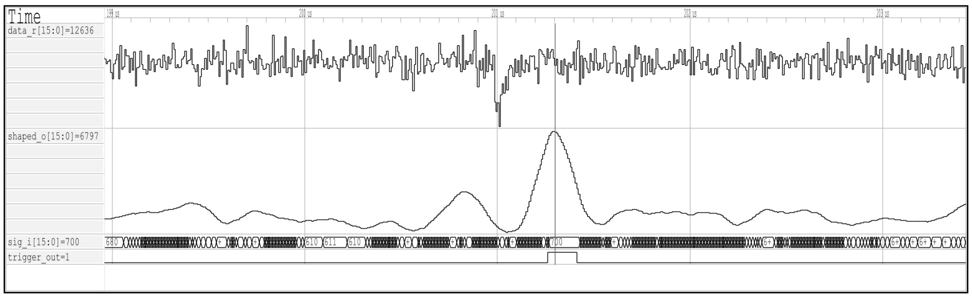


FIGURE 4.35: *Trigger output of the algorithm, "trigger_o", from the input "data_r". Here we can see that the pulse is discriminated with a confidence level of almost 10σ, having the top of the triangle, "shaped_o", a value of above 6700 and the standard deviation of the noise, "sig_i", a value of 700.*

Chapter 5

# Trigger Testing Results

The aim of this section is to focus on the testing of the algorithm to verify whether it has been successfully implemented in the FPGA and if its usage would bring advantages to the discriminations of low-amplitude signal pulses.
Two different tests have been performed:

- a verification of the relationship between the energy associated to the incoming pulses and the output triangle coming from the trigger shaping module
- a study of the discrimination capability of our trigger compared to a single leading-edge threshold trigger algorithm.

## 5.1 Trigger energy dependence

Tests have been performed on a GCU board with an ADU mezzanine board connected via FMC. The power board has not been used: a dedicated 12V and 5.8V have been provide to the GCU and ADU, respectively, using an external power supply module. A Digilent JTAG-HS3 Programming Cable provided a connection between the FPGA and the computer.
The input raw data has been provided by the CAEN DT5810D Dual Fast Digital Detector Emulator. The pulses are set to have a 0.01 $\mu$-seconds rise time, 0.05 $\mu$-seconds decay time and negative polarity, to best simulate a PMT output. All parameters including the pulse amplitude and frequency are controlled by the computer through the dedicated software "Detector emulator control center". To visualize the emulator output, a Tektronix TDS 2024B oscilloscope has been used.

The digital emulator used for the test is shown in Fig. 5.1.

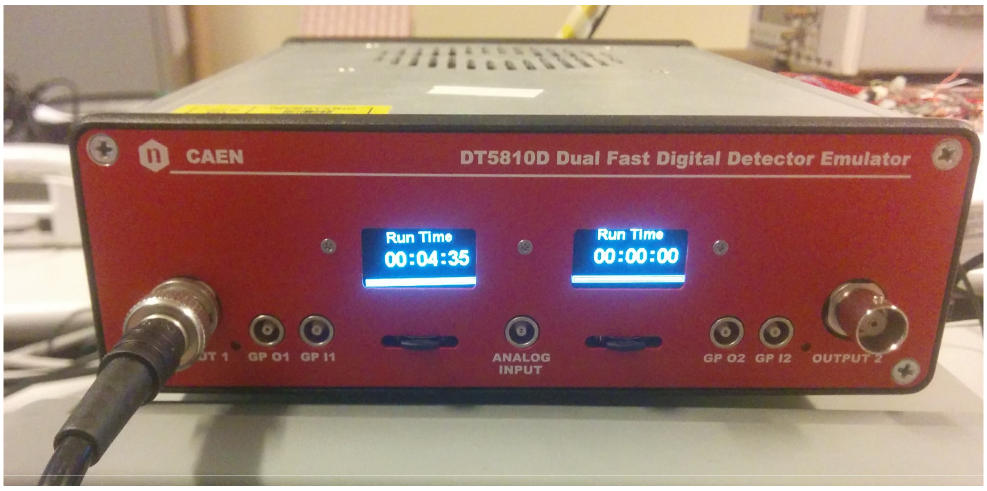


FIGURE 5.1: *CAEN DT5810D Dual Fast Digital Detector Emulator.*

For the whole duration of the test, the rise time and the decay constant of the pulses are fixed. This makes their energy directly proportional to the pulse amplitude.
A sample of triangle peak values measured as a function of the pulse voltage amplitude has been collected.
The pulse amplitude is measured with the oscilloscope, while the peak value is obtained through the Xilinx software Chipscope Pro. The resulted graph is shown in Fig. 5.2.

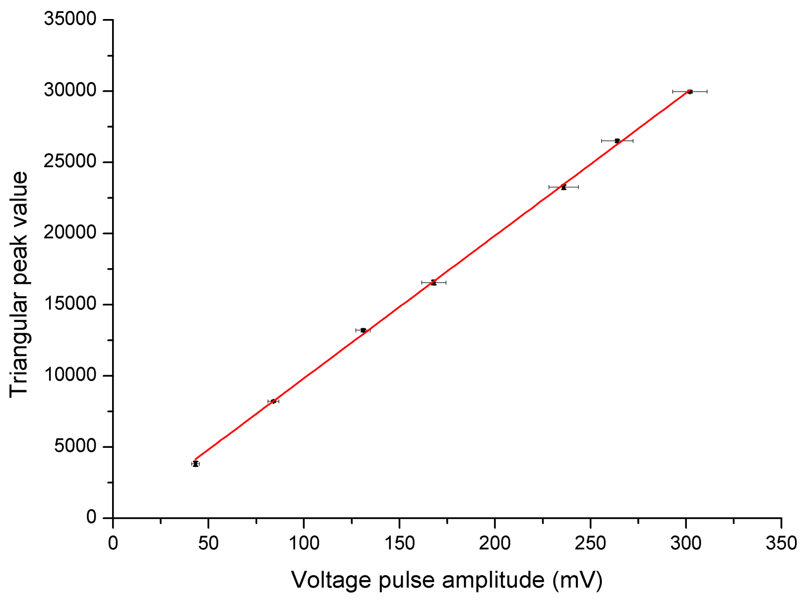


FIGURE 5.2: *Triangular peak value as a function of the voltage pulse amplitude. Error bars are present for both x-/y-axis values. The red line represents a linear fit to the sample.*

The error associated with the voltages is given by two contributions, summed in quadrature: the oscilloscope resolution and the scale factor. The resulted linear fit is

| Peak value $= a + b \cdot mV$ | |
|---|---|
| $a =$ | $-180 \pm 70$ |
| $b =$ | $(100.2 \pm 0.4)\, mV^{-1}$ |

TABLE 5.1: *Linear fit parameters.*

The compatibility of the linear fit intercept with the zero value is 2.6 .

The residual plot is shown in Fig. 5.3:

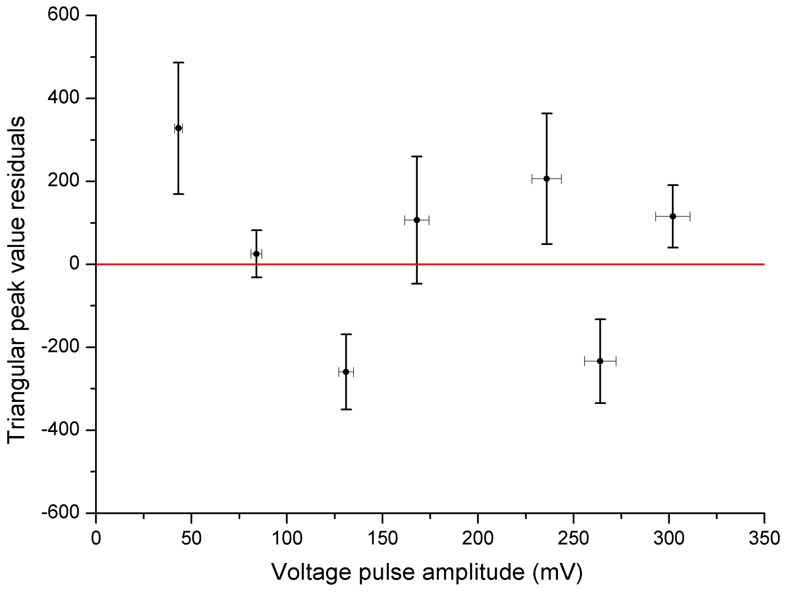


FIGURE 5.3: *Residual plot of the triangular peak values vs the pulse voltage amplitude.*

The residual plot shows a qualitative confirmation of the linear relationship between the triangle peak value and the pulse's voltage amplitude.
Since in this conditions the pulse's voltage amplitude is directly proportional to the energy of the pulse, the energy-peak proportion is verified.

## 5.2 Trigger efficiency

This test aims to study the discrimination capacity of the proposed algorithm.
A direct comparison with a leading-edge threshold trigger was performed under low-amplitude input pulses conditions. With controllable pulses available, a quantitative comparison of the discrimination capability of the trigger algorithm can be performed by an efficiency scan as a function of the pulse amplitude.
The efficiency can be defined as:

$$\textit{Efficiency} = \textit{Valid trigger requests} / \textit{Total trigger requests} \tag{5.1}$$

in fact, if the threshold level is too high, only a few pulses will be recognised, resulting in a low value of efficiency. A similar behaviour is observed if the threshold level is

too low, at the same level of background noise spikes, with the effect of raising the total trigger requests and lowering the efficiency. Similar results are expected in case of large amplitude pulses which are well above background noise; different behaviour is instead expected for lower amplitudes, close to the background level, resulting in different efficiencies for the MWD-MWA algorithm in comparison to the simple leading edge trigger.
The algorithm has been tested using the same **JUNO GCU board** connected to an ADU mezzanine board. The power distribution was provided by a JUNO power board, powered with 24 V from a power supply. The firmware of the GCU implements both the the MWD and MWA trigger algorithm and the the leading edge threshold trigger. Two Micro-Miniature CoaXial (MMCX) connectors are used as trigger output channels.

The signal source is a **20-inch Hamamatsu R12860-HQE PMT**, secured inside a dark-box, shown in Fig. 5.4. The Hammamatsu PMT socket, placed at the base of the PMT, has two BNC connections, one of high voltage, connected to an Ortec 556 High Voltage Power Supply, and a second one used for the output signal. During data acquisition the PMT output was fed to both the GCU board and to an oscilloscope for monitoring, the Teledyne LeCroy Waverunner 8254M-MS, through a resistive signal splitter.
The light inside of the dark-box was provided by a **Light Emitting Diode (LED)**. Its voltage supply was granted by a pulser with user settable parameters. To identify the LED light pulses, a trigger output channel was provided by the pulser.

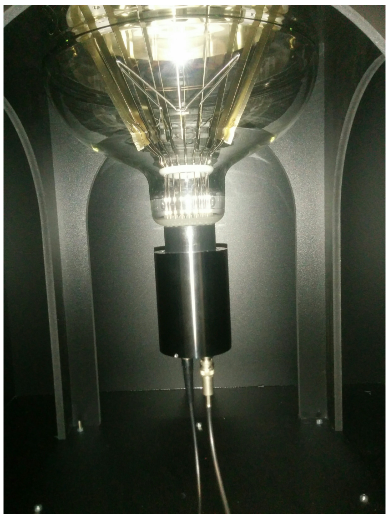


FIGURE 5.4: *JUNO large PMT used for the peak efficiency test inside the dark-box. The Hammamatsu socket is also shown.*

The use of a LED for the emission of a light signal has some peculiar characteristics. The most important, that may have an influence to the measurements, is that the LED is an incoherent light source. This means that the LED's pulse intensity follows a Gaussian distribution, keeping the supply voltage fixed. This effect can be seen in the pulse amplitude of the electrical output signals read by the GCU, which will follow the same statistics as well. This aspect of the LED light has relatively greater effects when the LED is supplied with low voltages.
As a consequence, a scan based on the amplitude of the pulse would be very complex. To simplify the measurement, a scan on the voltage supplied to the LED is carried out instead. Sufficient statistic was collected to minimize any effects given by the variability of the pulse amplitude.
It is now necessary to give a practical definition for *valid trigger requests*. For our purposes, it corresponds to a trigger signal generated by the LED signal. The trigger request is considered *valid* when it coincides with the external trigger generated by the pulser, within a predefined time window. The total trigger requests will be generated by several causes:

- LED light, which generates the *valid trigger requests*.
- Dark noise: the PMT has an intrinsic dark rate value, dependent on the temperature and on the supplied voltage. According to the PMT datasheet it is 18.5

kHz at 1800 V.

- Photons penetrated in the dark-box.
- In case of a low threshold level, electronics background noise.

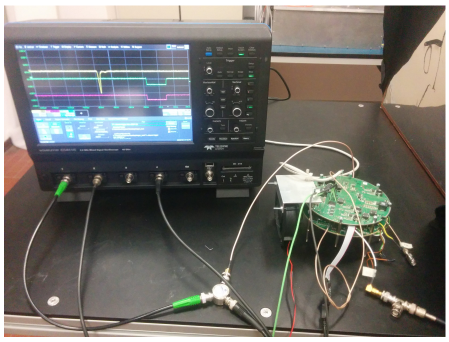


FIGURE 5.5: *GCU board assembled with the powerboard (right) and the Teledyne LeCroy Waverunner 8254M-MS (left) used to monitor the PMT output.*

FIGURE 5.6: *Power supplies: on top power board/GCU power supply; on bottom the pulser used for the LED light. Only one channel has been used.*

Fig. 5.7 defines the analogue acquisition chain that has been used to acquire the efficiency measures. The number of trigger requests sent by the GCU during a pre-defined acquisition time window, are counted by the CAEN Mod. N145 [32] Quad Scaler and Preset Counter-timer. The module presents a NIM logic input channel, which increments the counts for every falling edge of the signal (the NIM logic level 1 has a lower voltage than the NIM logic level 0) and a gate input channel which enables the counting, constraining the acquisition.
To obtain *total trigger requests*, unconstrained acquisitions of GCU trigger output requests are performed, enabling the counting for a time window of 60 seconds. Since the counter accepts NIM logic inputs, a CAEN Mod. N413 [33] leading-edge discriminator has been used.
To obtain *valid trigger requests*, the acquisition window is enabled by the pulser trigger output in order to register a trigger request only when caused by the LED light. The computational delay of the GCU trigger output must be taken into consideration, therefore an Ortec GG8010 [34] Octal gate generator has been used to add a delay to the pulser trigger output, followed by a TTL to NIM converter. The acquisition time window is still set at 60 seconds.

The modules used for the test and a *valid trigger request* waveform are showed in Fig. 5.8 and in Fig. 5.9 respectively.

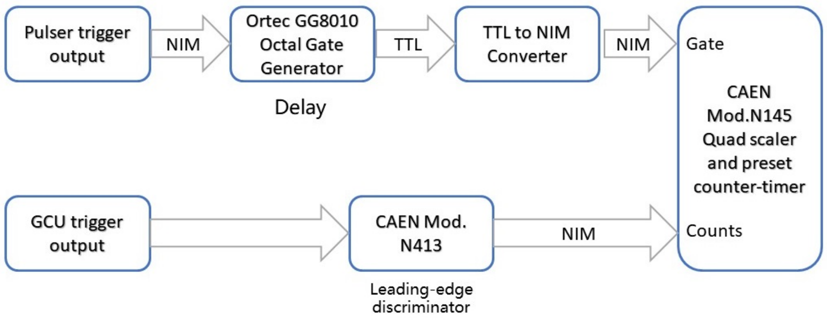

Pulser trigger output
NIM
Ortec GG8010 Octal Gate Generator
Delay
TTL
TTL to NIM Converter
NIM
Gate
CAEN Mod.N145 Quad scaler and preset counter-timer
Counts
GCU trigger output
CAEN Mod. N413
Leading-edge discriminator
NIM

FIGURE 5.7: *Analogue acquisition chain used for the trigger efficiency test.*

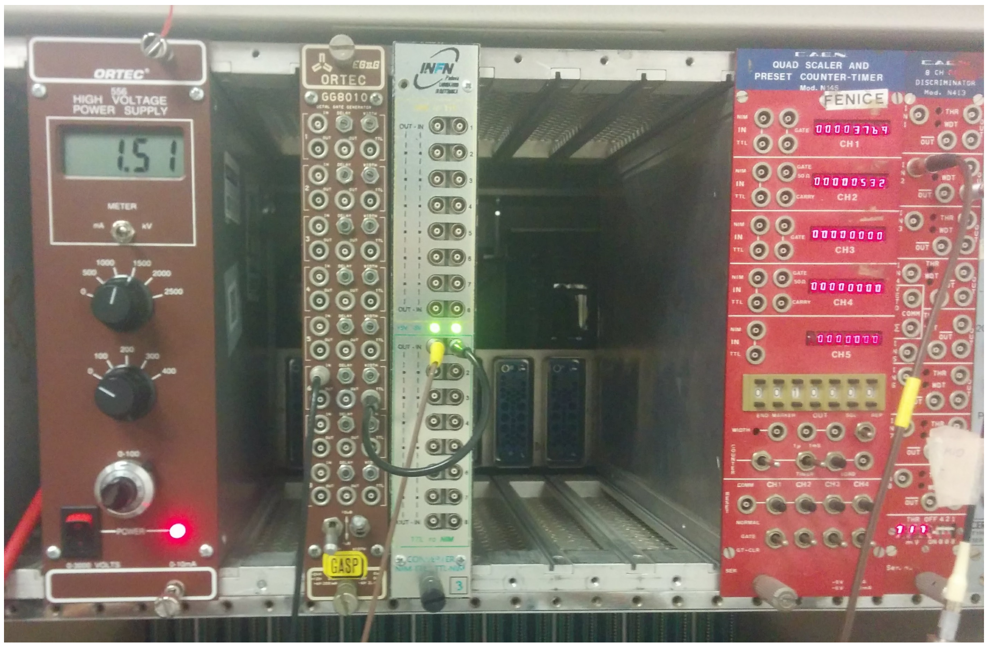


FIGURE 5.8: *NIM modules used for the efficiency test. From left to right: High Voltage power supply, Gate generator, TTL to NIM module, Counter, Discriminator.*

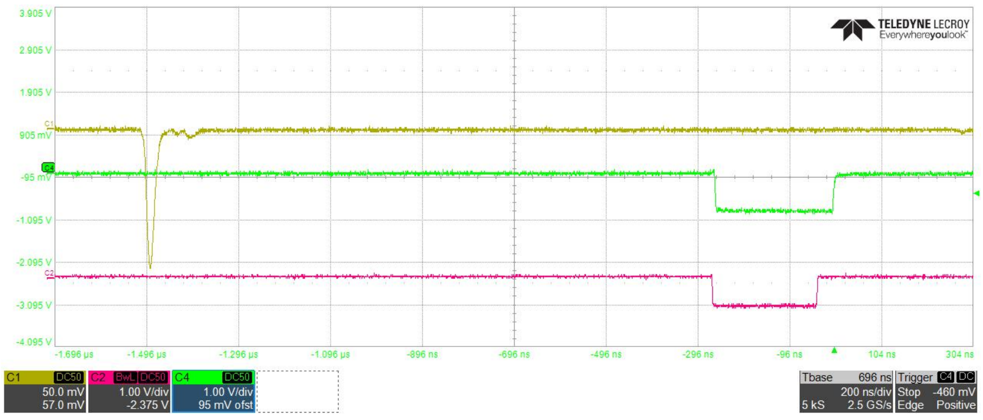


FIGURE 5.9: *Waveform of a valid trigger request. The top yellow plot is the PMT output, the middle green plot is the gate NIM signal given by the pulser trigger output and the purple bottom plot is the GCU trigger NIM output. The waveform is obtained by the LeCroy Waverunner 8254M-MS oscilloscope.*

### 5.2.1 Data analysis and results

The first operation has been to estimate the average amplitude of the pulses as a function of the LED supplied voltage. This serves to give an idea on what pulse amplitude we will discriminate for the efficiency calculation. As shown in Fig. 5.10, the PMT's output pulse amplitude caused by the LED light follows a Gaussian distribution. The plot representing the pulse amplitude as a function of the LED supply voltage is shown in Fig. 5.11:

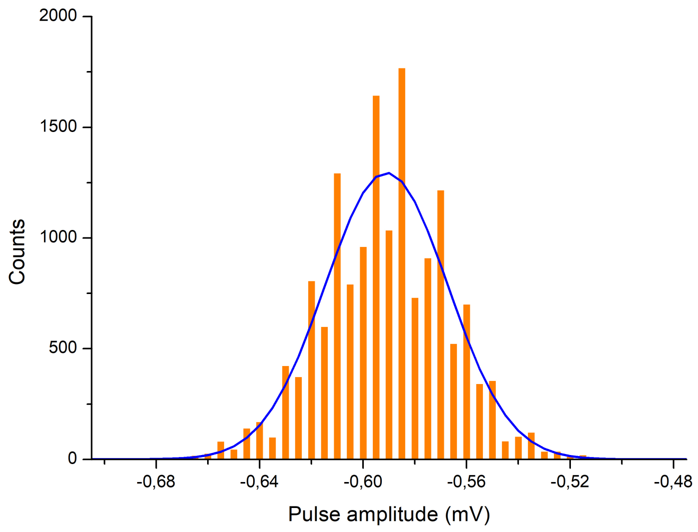


FIGURE 5.10: *Gaussian distribution of the PMT's output pulse peak voltage corresponding to a LED supply voltage of 3.4 V*

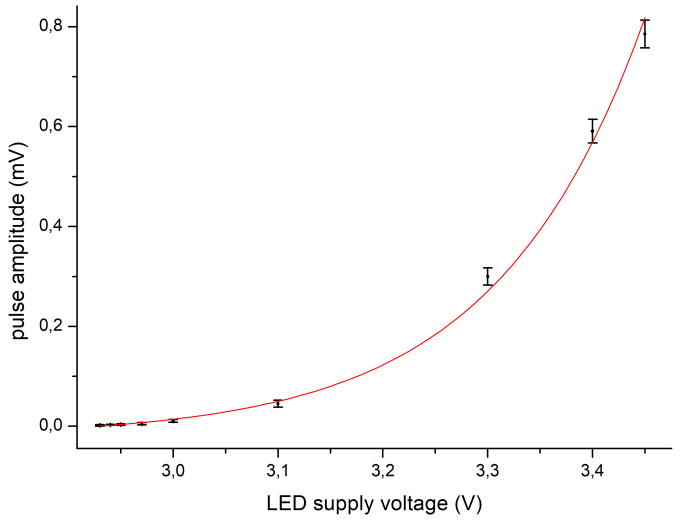


FIGURE 5.11: *Pulse amplitude as a function of the LED supply voltages. The error bars on the plot corresponds to the sigma of the Gaussian distribution. The red line is the result of an exponential fit.*

As shown, the pulse's amplitude have an exponential relationship with the voltages supplied to the LED. The exponential fit parameters are:

| $Pulse\ amplitude = a * e^{-V/b} + c$ | |
|---|---|
| $a =$ | $(3 \times 10^{-11} \pm 4 \times 10^{-11})\, mV$ |
| $b =$ | $(-0.14 \pm 0.01)\, V^{-1}$ |
| $c =$ | $(-0.022 \pm 0.006)\, mV$ |

TABLE 5.2: *Exponential fit parameters.*

For every led voltage value, an efficiency scan for both algorithms is performed as a function of the threshold level. Only the highest efficiency values are kept and plotted. Each scan is performed for the same time interval; since the timer embedded in the counter was not available, a manual chronometer was used, choosing 60 seconds as the time interval.

The resulting plot is shown in Fig. 5.12:

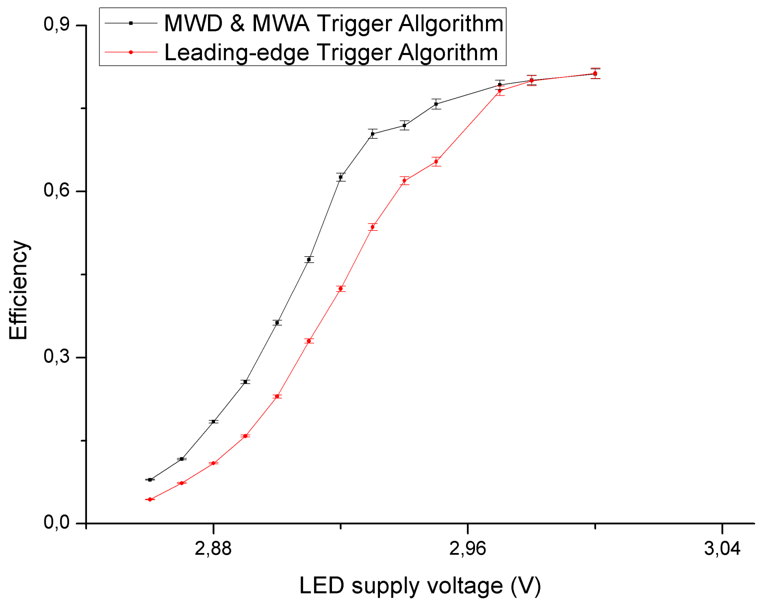


FIGURE 5.12: *Efficiency for each led supply voltage. The MWD-MWA trigger algorithm is rapresented by the black points, while the simple leading-edge trigger is shown with the red points.*

As expected, for voltages around 3 V, which consists in pulses well distinguishable from the background noise, the discrimination efficiency is the same for both trigger algorithms. As we lower the led supply voltage, the MWD-MWA trigger algorithm outperforms the leading-edge algorithm.
The error associated to the efficiency values is obtained by the propagation of uncertainty formula from the errors on the counts, which is obtained by the formula:

$$\sigma_{\text{counts}} = f_{\text{counts}} * \sigma_{\text{t acq.}} \tag{5.2}$$

where $f_{\text{counts}} = counts/t_{\text{acq.}}$, with $t_{\text{acq.}}$ being the acquisition time, represents the average number of counts per seconds, and $\sigma_{\text{t acq.}}$ represents the estimated error on the acquisition time, chosen to be 0.5 seconds.

Another significant result is obtained by analyzing the plot in Fig. 5.13:

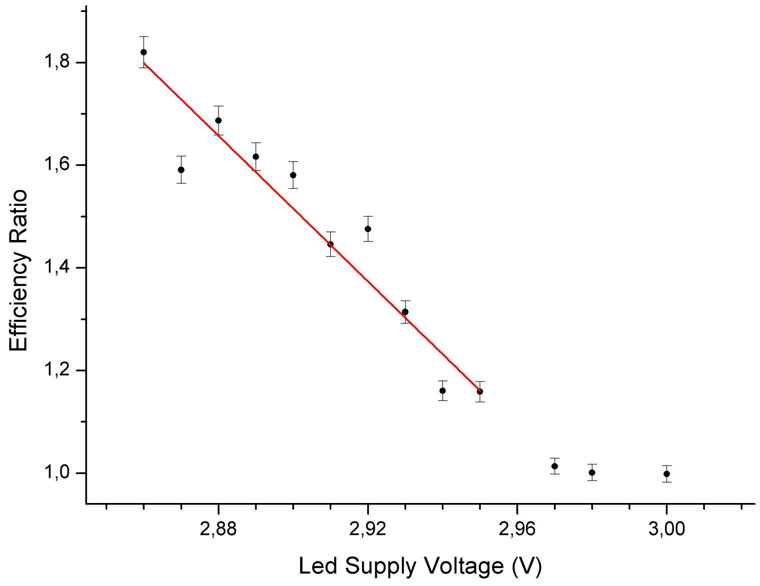


FIGURE 5.13: *Efficiency$_{(MWD-MWA\ trigger)}$/Efficiency$_{(leading\text{-}edge\ trigger)}$ as a function of the led supply voltage. The red line represents the linear fit.*

Plotting the ratio of the efficiencies, a linear relationship starting from the LED supply voltage of 2.95 V is perceived, underlining the advantages of the trigger presented in this thesis for low amplitude pulses.
The results of the linear fit is:

| *Efficiency ratio* $= a + b \cdot V$ | |
|---|---|
| $a =$ | $22.1 \pm 0.8$ |
| $b =$ | $(-7.1 \pm 0.3)\, V^{-1}$ |

TABLE 5.3: *Linear fit parameters.*

To verify the correctess of the measurements, we have to check that, with the LED disabled (but still enabling the pulser trigger output) no gated counts are registered, leaving this sort of dark efficiency close to the zero value. The test was performed at the lowest threshold levels used during testing for both the algorithms, and with an acquisition time of 60 seconds. The results are the following:

| LED Supply Voltage | MWD-MWA trigger alg. | | | Leading-edge trigger alg. | | |
|---|---|---|---|---|---|---|
| V | Valid trg | Total trg | Efficiency | Valid trg | Total trg | Efficiency |
| 2.86 | 4 | 54'096 | $7 \times 10^{-5}$ | 18 | 26'214 | $7 \times 10^{-4}$ |

TABLE 5.4: *Dark efficiency measures.*

For such small values of *valid trigger requests* it's not possible to compute an error using Eq. 5.2, as the hypothesis of a uniform distribution of the counts during the time interval, needed to find the counting frequency, is no more legitimate. Nevertheless, the efficiency values are so small, that is safe to say that the dark efficiency can be considered negligible.

Chapter 6

# Conclusion

The MWD-MWA trigger algorithm has been successfully implemented into the FPGA and the test performed with a real PMT demonstrate an effective improvement in the discrimination capacity of low-amplitude signals.
Measures obtained using an LED light source, with controllable supply voltage, show a logarithmic relationship for low-amplitude pulses between the efficiency ratio of the MWD-MWA algorithm to a leading-edge algorithm, and the average pulse amplitude registered by the PMT, shown in Fig. 6.1:

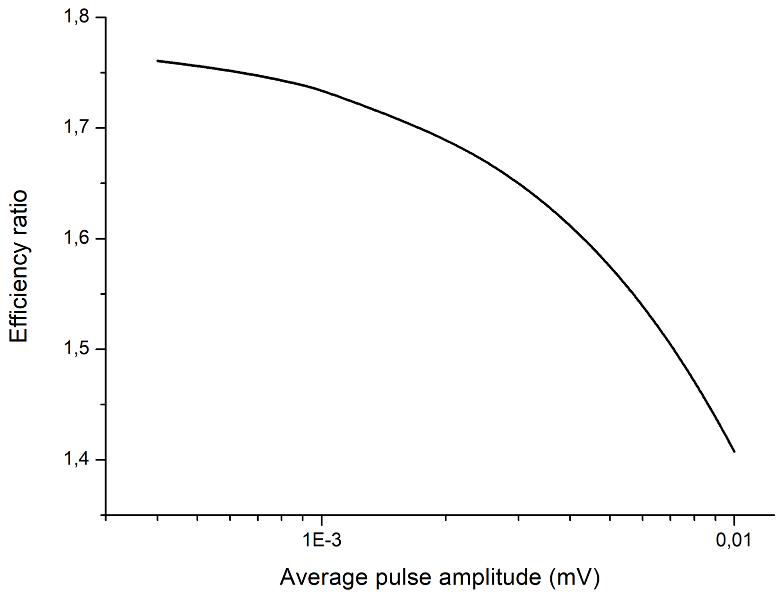


FIGURE 6.1: *Relationship between the Efficiency ratio of the tested trigger algorithms and the average pulse amplitude obtained by the Hamamatsu PMT using a LED light source.*

| *Efficiency ratio* $= a + b \cdot \ln(mV + c)$ | |
|---|---|
| $a =$ | $-2 \pm 2$ |
| $b =$ | $-9.9 \times 10^{-1} \pm 8 \times 10^{-2}$ |
| $c =$ | $0.022 \pm 0.006$ |

TABLE 6.1: *Logarithmic parameters of Fig. 6.1 obtained from Tab. 5.2 and Tab. 5.3. The errors are calculated using the propagation of uncertainty formula.*

In particular, signals produced in response to the light emitted by the LED powered at 2.86V, (this voltage corresponds to an average pulse amplitude of about 350 $\mu$V according to calibration) show an improvement of the efficiency of the MWD-MWA algorithm compared to the simple leading edge trigger of a factor $1.82 \pm 0.03$. Further tests will be performed with a coherent light source, in order to have a well-defined amplitude of the pulses at the PMT output; this source can be a pulsed laser. Our aim is to exploit the trigger algorithm hereby described, together with a coherent light source and a pulse amplitude energy calibration procedure to discriminate and trigger on a single p.e. signal. To further increase the efficiency and the discrimination capacity, the digital implementation may be parallelized in order to exploit all the samples instead of 1 out of 4. This optimization will exclude signal decimation and loss of information.

# Appendix

## Text file in VHDL simulation

The simulations performed in Sec. 4.3, import a text file to the VHDL test bench. The part of the code that allows it, is attached below:

```vhdl
USE ieee.std_logic_1164.ALL;
USE ieee.std_logic_textio.all;
USE std.textio.all;
USE ieee.numeric_std.ALL;

COMPONENT baseline_follower
    PORT(
          data_from_text : IN  std_logic_vector(15 downto 0);
          ...
         );
    END COMPONENT;


--Inputs
   signal data_from_text : std_logic_vector(15 downto 0) := (others =>
   '0');

   BEGIN
  uut: module_name PORT MAP (
           data_from_text => data_from_text,
           ...
           );

process
   FILE f : TEXT;
   constant filename : string :="file_name.txt";
   variable l : LINE;
   variable i : integer:=0;
   variable b : std_logic_vector(15 downto 0);
  File_Open (f,filename, read_mode);
  while ((i<=35000) and (not EndFile (f))) loop
    readline (f, l);
    read(l, b);
    data_from_text <= b;
    i := i + 1;
    wait for clk_period;
  end loop;
  File_Close (f);
  wait;
end process;
```

# Bibliography

[1] F. P. An et al., *Measurement of electron antineutrino oscillation based on 1230 days of operation of the Daya Bay experiment*, Phys Rev D95(2017)072006 arXiv:1610.04802

[2] S. H. Seo et al, *Spectral Measurement of the Electron Antineutrino Oscillation Amplitude and Frequency using 500 Live Days of RENO Data*, arXiv:1610.04326

[3] Y. Abe at al, *Measurement of q13 in Double Chooz using neutron captures on hydrogen with novel background rejection techniques*, JHEP01 (2016) 163, arXiv:1510.08937

[4] F. Capozzi et al, *Global constraints on absolute neutrino masses and their ordering*, Phys Rev D 95 (2017), arXiv:1703.04471

[5] P. Ghoshal and S.T.Petcov, *Neutrino mass hierarchy determination using reactor antineutrinos*, JHEP 1103:058, 2011, arXiv:1011.1646

[6] F.P. An et al, *Neutrino Physics with JUNO*, J. Phys. G43 (2016) 030401, arXiv:1507.05613

[7] G. Ranucci et al, *Status and prospects of the JUNO experiment*, Neutrino 2016 Conference, London July 4 - 9 2016

[8] Steven Lau, *CPR1000 Design, Safety Performance and Operability* Daya Bay Nuclear Power Operations and Management Company, 5 July 2011

[9] T. Adam et al, *JUNO Conceptual Design Report*, arXiv:1508.07166

[10] D. Pedretti et al, *The Global Control Unit for the JUNO front-end electronics*, Proceeding of TIPP 2017, to appear in Springer Proceedings in Physics

[11] Wikipedia. Field Programmable Gate Array, *en.wikipedia.org/wiki/Field_Programmable_Gate_Array.*

[12] A. Olshevskiy, *HV Unit Report*, JUNO internal report

[13] Hu Jun et al, *ADU(v1.4) Performance Test*, JUNO internal report

[14] Stoica, V. Ionut. *Digital pulse-shape analysis and controls for advanced detector systems*, Groningen : s.n., 2012. 129 p.

[15] Fabio Rocca (Dipartimento di Ingegneria Elettronica e Informazione Politecnico di Milano), *Elaborazione Numerica dei Segnali*, 2010

[16] C. Ghabrous Larrea et al,*IPbus: a flexible Ethernet-based control system for xTCA hardware*, JINST 10 (2015), C02019

[17] Timing, Trigger and Control (TTC) Systems for the LHC, *http://ttc.web.cern.ch/ttc/*

[18] ISE Design Suite, *https://www.xilinx.com/products/design-tools/ise-design-suite.html*

[19] Xilinx, Inc, *7 Series FPGAs SelectIO Resources, User Guide*, UG471 (v1.9), August 22, 2017

[20] Xilinx, Inc, *7 Series FPGAs Clocking Resources, User guide*, UG472 (v1.13), March 1, 2017

[21] Xilinx, Inc, *LogiCORE IP FIR Compiler v6.3, User guide*, DS795, October 19, 2011

[22] Xilinx, Inc, *LogiCORE IP FIFO Generator v9.3, Product guide*, PG057, December 18, 2012

[23] A. Georgiev, W. Gast, and R.M. Lieder, *An analog-to-digital conversion based on a moving window deconvolution*, IEEE Trans. Nucl. Sci., 41:1116, 1994.

[24] M. Kavatsyuk et al, *Performance of the prototype of the electromagnetic calorimeter for PANDA*, Nucl. Instr. Meth. A, 2011.

[25] Fixed-Point Designer, *https://it.mathworks.com/products/fixed-point-designer.html*

[26] Filter Design, *https://it.mathworks.com/discovery/filter-design.html*

[27] GTKWave, *http://gtkwave.sourceforge.net/*

[28] Q. Liu et al, *Nucl. Instrum. Meth. A 795, 2015, 284*, arXiv:1504.01001 [physics.ins-det], 2015

[29] W. Konetschny and W. Kummer, *Phys. Lett. B 70*, 1977, 433.

[30] E. Tuzza, *Studio della risposta temporale di fotomoltiplicatori di grandi dimensioni per esperimenti sulla fisica del neutrino*, A.A. 2015/2016, Laurea in Fisica

[31] Xiang-Cui Lei et al, *Evaluation of new large area PMT with high quantum efficiency*, Chinese Physics C 40 (2016), 26002

[32] CAEN Mod. N145, Datasheet: *http://www.caen.it/servlet/checkCaenManualFile?Id=5256*

[33] CAEN Mod. N413, Datasheet: *http://www.caen.it/servlet/checkCaenManualFile?Id=5198*

[34] Ortec GG8010, *http://ortec-online.com/products/electronics/delays-gates-and-logic-modules/gg8020*