THE PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA

MINISTRY FOR HIGHER EDUCATION AND SCIENTIFIC RESEARCH

IBN KHALDOUN UNIVERSITY -TIARET-

FACULTY OF APPLIED SCIENCES

ELECTRICAL ENGINEERING DEPARTMENT

GRADUATION MEMOIR

To obtain The Master's Degree

Field : Science & Technology
Sector : Electrical Engineering
Specialty : Electrical Control

MEMOIR THEME

Numerical Implementation of The Analysis and Maximas Detection Function of a Shaped Temporal Signal in Gamma-Ray (γ) Spectroscopy Chain

Prepared By:

Mr. AYAD AHMED DJAMEL
Email : ahmeddjamelayad@gmail.com

Mr. MESSLEM ABDELKADER
Email : abdelkadermesslem99@gmail.com

Presented in front of the jury composed of:

Dr. A. GOUICHICHE	MCA: Tiaret University	President
Dr. R. OTHMANI	MAA: Tiaret University	Examiner
Dr. A. BENATTIA	MCB: Tiaret University	Examiner
Dr. A. MESSAI	Research Director: CRNB	Supervisor
Dr. A. SAFA	MCB: Tiaret University	Co-Supervisor

2021 - 2022

This Project is a Collaboration Between Center of Nuclear Research of Birine -Djelfa, Algeria- (CRNB) & Laboratory of Electrical Engineering and Plasma -Tiaret, Algeria- (LGEP)

Acknowledgements

We thank the Almighty Allah for the help and health and strength to start and finish this work. We also thank our parents and family, for supporting us during all those years to this point, without them we won’t be here.

We would like to express our special thanks of gratitude to Mr. MESSAI ADDNANE, for the quality of this exceptional mentoring, patience and his availability during the preparation of this thesis, he gave us the opportunity to do this wonderful project and provide us with all the necessary support, information and hardware components. We are also thankful to Mr. SAFA AHMED for being patience with us, guiding us, supervising us and following us during this work. We also want to thank Mr. KOUDIAH NOUREDDINE for welcoming and sharing his time to support and teach us.

Besides, we would like to give all thanks to LGEP laboratory members for sharing their space of work for us and for all the mental support. Our heartfelt thanks also go to the members of the jury who agreed to read and evaluate this work.

Finally, we would like to thank all Professors for their encouragement and all people who support us in our project.

Table of Contents

Figures List

Chapter I

Chapter II

Chapter III

Chapter IV

Table List

Chapter I

Chapter II

Chapter III

Chapter IV

Abbreviations List

Abbreviations	**Meaning**
HPGe	**H**igh **P**urity **Ge**rmanium
ADC	**A**nalog to **D**igital **C**onverter
DPP	**D**igital **P**ulse **P**rocessing
MCA	**M**ulti**c**hannel **A**nalyzer
CSP	**C**harge-**S**ensitive **P**reamplifier
DTS	**D**igital **T**rapezoidal **S**haping
SNR	**S**ignal-to-**N**oise **R**atio
MSB	**M**ost **S**ignificant **B**it
RAM	**R**andom **A**ccess **M**emory
DAC	**D**igital to **A**nalog **C**onverter
FPGA	**F**ield **P**rogrammable **G**ate **A**rray
PL	**P**rogrammable **L**ogic
PS	**P**rocessing **S**ystem
VHDL	**V**ery High-Speed Integrated Circuits **H**ardware **D**escriptive **L**anguage
SoC	**S**ystem **o**n **S**hip
EPP	**E**xtensible **P**rocessing **P**latform
PC	**P**ersonal **C**omputer
AMBA	**A**dvanced **M**icrocontroller **B**us **A**rchitecture
IC	**I**ntegrated **C**ircuit
BD	**B**lock **D**esign
IP	**I**ntellectual **P**roperty
SDK	**S**oftware **D**evelopment **K**it
AXI	**A**dvanced e**X**tensible **I**nterface
COM	**COM**munication Port
TCL	**T**ool **C**ommand **L**anguage
OS	**O**perating **S**ystem

General Introduction

The gamma ray spectroscopy system is an essential tool in radiometric assay; it allows identification and quantification of the radio nuclides of gamma emitters present in gamma sources such as the nuclear industry, geochemical investigation, and astrophysics. The analysis of the resulting gamma ray energy spectrum in a gamma ray spectrometer is a representation of the distribution of the intensity of gamma radiation versus the energy of each gamma photon emitted.

Gamma spectroscopy is the study and analysis of gamma ray Spectrometers. Gamma ray spectrometers are the tools that monitor and collect such data as it detects an incident gamma photon by a High Purity Germanium (detector that produces a low amplitude, short duration current pulse which needs to be converted into a sufficient duration pulse that is easy to analyze using a preamplifier. However, there are various noise sources and errors in the system that must be eliminated or reduced. A specific digital shaper is used to increase the resolution of the displayed spectrum; thus, measure the energy of the detected gamma photon with a higher precision.

Similar to many scientific concepts, spectroscopy was developed by the cumulative work of many scientists over many years. Two architectures of the spectroscopy systems were used to analyze electromagnetic gamma radiation from various nuclear reactions: analog spectroscopy systems that were used in early stages and digital spectroscopy systems. Now days, with the emerging of the fast analog to digital converters and the programmable logics, digital spectroscopy system works similarly, if not better, to analog spectroscopy as it has the ability to implement digital filters that mostly cannot be implemented by analog systems. On which, the digital system is the architecture used in this thesis that is divided into four chapters:

- In chapter one, we discussed about the physics behind the electromagnetic waves in general and in nuclear reactions particularly, focusing on gamma rays, its detection technique, and the concept of spectroscopy systems. We then gave an overview of the basics of gamma ray spectroscopy, the structure and its different stages, and how they perform together to measure the energy of the detected gamma photons. We also mentioned the sources of error in the spectroscopy chain and the noise that affect the measurement efficiency.
- In chapter two, we detailed the multichannel analyzer which is the heart of our work, starting with the trapezoid filter and how it was developed by deriving it mathematically and showing the resolution improvements when using it by reducing the measurements

error that occur. We also put the scope on the peak detection algorithm and the classification of the maxima values in order to plot the spectrum.

- In chapter three, we simulate the spectroscopy system under MATLAB using Simulink and Xilinx System Generator blocks. Xilinx blocks are used to generate a VHDL code that is used later in the implementation phase. We also deeply analyze the quantization error of the trapezoidal filter in order to choose the optimal parameters of the Xilinx blocks which would be implemented in the hardware design.
- In chapter four, we talked about the experimental part based on the Zedboard System-on-Chip. We describe the used hardware, the different components, and the technique used to generate the FPGA implementation from the system generator. Using Vivado Design Suite, IP Integrator and Software Development Kit (SDK) we were able to perform the multichannel analyzer and connect it to software that we developed to plot the spectrum in real-time.

Chapter I

Generality about Gamma Ray Spectroscopy Systems

I.1 Introduction

In this chapter, we present the study of gamma photons energy and the basics of spectroscopy, with an emphasis on system architecture. In general, spectroscopy is the study of emission of electromagnetic radiations and involves presenting it in a spectrum which makes it a vague concept. However, in gamma ray spectroscopy, we measure the number of gamma photons emitted according to their individual energy in order to display the spectrum by passing it through a number of stages that is described in this part.

I.2 The Physic of Gamma rays

Faraday's equations of the phenomenon of electromagnetic induction concludes that a time changing of magnetic field at a point produces an electric field at that point. Maxwell pointed out that there is symmetry in nature, therefore, when the electric field changes over time at a point it produces a magnetic field at that point. This means that a change in one field with time (whether it is electric or magnetic) produces another field. This idea led Maxwell to conclude that variations in electric and magnetic fields are perpendicular to each other and result in electromagnetic disturbances in space. These disturbances have wave properties and propagate through space without any physical medium, they are called electromagnetic waves and they can travel in vacuum or free space at a speed of $c = 3 \times 10^8\ m.s^{-1}$ [1].

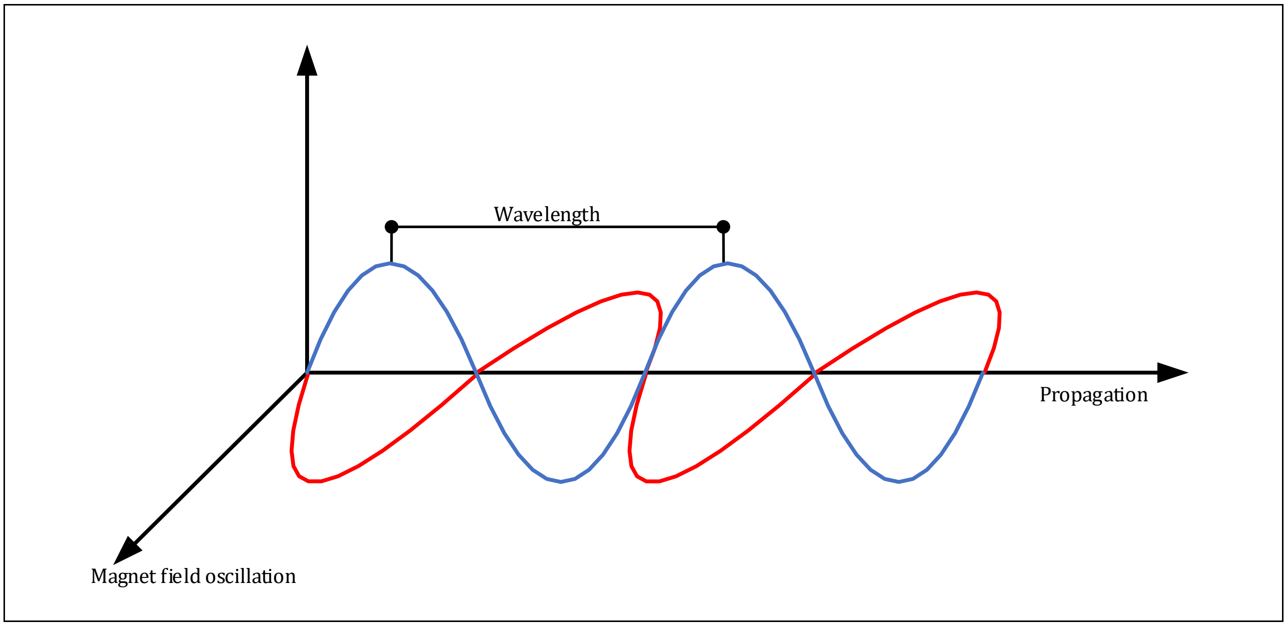


Figure I.1 The propagation of the electromagnetic wave.

An electromagnetic wave in a vacuum consists of perpendicular electric and magnetic fields that oscillates alternately making it a transverse wave, because the fields are perpendicular to the direction in which the wave is traveling.

The electromagnetic waves are characterized by the wave length λ which is the distance between the individual waves (e.g., from one peak to another) and the frequency **ν**, represent the number of waves which pass through a point in space each second. It is related to the velocity of the corresponding wave in a vacuum by the relation $c = \nu\lambda$.

Various phenomena are actually electromagnetic waves (radiation) of different wavelengths, energies and frequencies such as: radio, television, microwave, infrared, light, ultraviolet, X-ray, and Gamma rays. All these types of radiation make up the electromagnetic spectrum shown in Fig 2.I

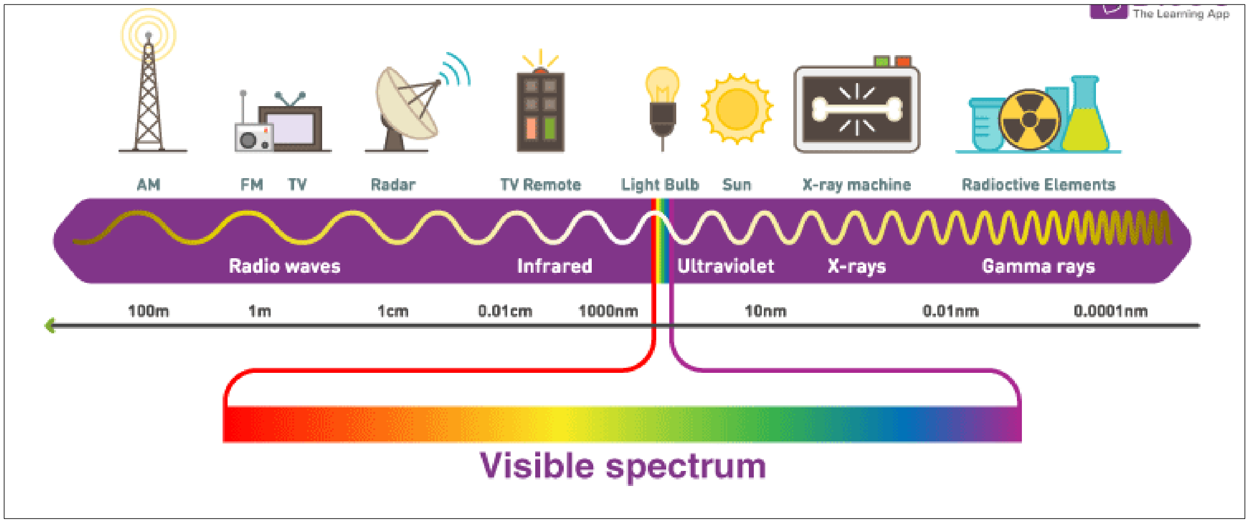


Figure I.2 The electromagnetic spectrum [2].

It should be pointed out that just like mechanical waves carry energy, electromagnetic waves also carry energy through space. In fact, an electromagnetic radiation is a bundle of energy contained in its electric and magnetic field that is called photons when viewed as particles. The electromagnetic radiation energy has difficult characteristic to describe, and can be viewed as it has wavelike and particle like properties that is represented by a photon, having no electric charge or rest mass and one unit of spin; they are field particles that are thought to be the carriers of the electromagnetic field [3].

The energy of one photon is given by Planck formula:

$$E = h\nu = \frac{hc}{\lambda} \qquad \text{I.1}$$

h: Planck constant, $\mathbf{h = 6.62607004 \times 10^{-34} \frac{m^2Kg}{s}}$**.**

ν: The corresponding electromagnetic wave frequency.

λ: wavelength.

For example, the energy of one gamma ray photon that have frequency $\nu = 5.92 \times 10^{20}$Hz is:

$$E = h\nu = 6.62607004 \times 10^{-34} \times 5.92 \times 10^{20} = 2.45\text{Mev} \qquad \text{I.2}$$

In the context of spectrometry which is the subject of our study, we will focus on the Gamma electromagnetic radiation.

Gamma rays have the smallest wavelengths and the most energy of any wave in the electromagnetic spectrum. They are produced by the hottest and most energetic objects in the universe, such as neutron stars and pulsars, supernova explosions, and regions around black holes. On Earth, gamma waves are generated by nuclear explosions, lightning, and the less dramatic activity of radioactive decay [4].

I.3 Gamma rays in nuclear reaction

Nuclear reaction in nature or artificial generally produces electromagnetic gamma radiation.

I.3.1 Spontaneous decay

The decay of a radioactive element into a stable or radioactive nucleus is accompanied by a radiation emission. This radiation emission consists of particulate radiation that can be charged or neutral and electromagnetic radiation.

- **Alpha radiation (α particle):** corresponds to the emission of a helium nucleus made up of 2 protons and 2 neutrons. It generally concerns heavy radionuclides.
- **Beta radiation (β):** concerns unstable nuclei rich in protons or neutrons. It has two forms: β^+ and β^-

  - $\boldsymbol{\beta^{-}}$: is a negatively charged electron associated with a particle called Antineutrino results from the transformation of a neutron into a proton. It concerns nuclei having an excess of neutrons.
  - $\boldsymbol{\beta^{+}}$: is a positively charged electron (e^{+}) called a Positron, and associated with that of a Neutrino (nearly zero mass).
- **Gamma radiation (γ):** is an electromagnetic radiation emitted as a result of radiation α or β, when these produce a new nucleus in excited state that decays by gamma-ray emission. Note that an alpha or beta decay of a given nuclear species is not always accompanied by gamma-ray emission and depends on the state of the daughter nucleus and the decay way in case of excited state of the daughter nucleus that can be either gamma decay or internal conversion. This excess energy is released in the form of high-energy photons with a high potential for penetration into matter. The energy of each photon can be calculated as shown in Eq I.3

$$\begin{gathered} {}^{233}_{92}\mathrm{U} \to {}^{229}_{90}\mathrm{Th}^{*} + {}^{4}_{2}\mathrm{He} \\ {}^{229}_{90}\mathrm{Th}^{*} \to {}^{229}_{90}\mathrm{Th} + \gamma \end{gathered} \qquad \text{I.3}$$

I.3.2 Fusion reaction

Fusion is the process where two light nuclei combine together releasing vast amounts of energy, for example:

$${}^{1}_{1}\mathrm{H} + {}^{2}_{1}\mathrm{H} \to {}^{3}_{1}\mathrm{He} + \gamma \qquad \text{I.4}$$

I.3.3 Fission reaction

Fission is the splitting of a heavy, unstable nucleus into two lighter nuclei releasing vast amounts of energy, for example:

$${}^{1}_{0}\mathrm{n} + {}^{235}_{92}\mathrm{U} \to {}^{141}_{56}\mathrm{Ba} + {}^{92}_{36}\mathrm{Kr} + 3\,{}^{1}_{0}\mathrm{n} + \gamma \qquad \text{I.5}$$

In both fission and fusion, large amounts of energy are given off in the form of heat, light, and gamma radiation on which the sum of the result fragments masses is less than the original mass. This missing mass **Δm** has been converted into energy according to Einstein's Eq I.6.

$$E = \Delta m \times c^2 \quad \text{I.6}$$

It should be pointed that the nuclear reactions are more complicated than this and detailing them is beyond the scope of this thesis.

I.4 Gamma-ray spectroscopy system

In this part we represent the basic concepts of the spectroscopy systems in general and gamma spectroscopy systems particularly.

The term spectroscopy is vague and refers generally to the analysis of data that is arranged along one or more axes where the graphical representation of this distribution is called the spectrum. Spectrum type requires specifying the nature of the data and the dimension of the axes (frequency in Hz, energy in eV or masse in Kg…etc.). In gamma spectroscopy, it is concerned with measuring the distribution of gamma photons according to their individual energy in eV, or a multiple of this unit.

Gamma rays from radio-nuclides are emitted with a rate and energy spectrum that is unique to the nuclear species. This uniqueness provides the basis for most gamma-ray assay techniques: Ideally, by counting the number of gamma rays emitted by a radioactive source or produced by a nuclear reaction with a specific energy which is the gamma ray energy spectrum, it is possible to determine the number of nuclei that emit and to characterize the composition, in terms of radio-nuclides present, and the activity, i.e. the number of disintegrations per second or Becquerels, of a source emitting gamma photons [5], [6].

For example, the final result of a gamma spectrometry measurement is statement of the flowing form: “the radioactive source contains Cesium 137 with an activity of 4500 Becquerels and Europium 152 with an activity of 3700 Becquerels”.

There are two kinds of energy spectra: The differential energy spectrum and the integral energy spectrum; however, for this work we will only be interested in differential energy spectra. That is a function n(E)dE such that n(E)dE is the number of particles whose energies are: $E < \text{energie of gamma particles} < E + dE$ [6]. Fig I.3 shows a simulated versus a real gamma ray energy spectrum of Cesium 123.

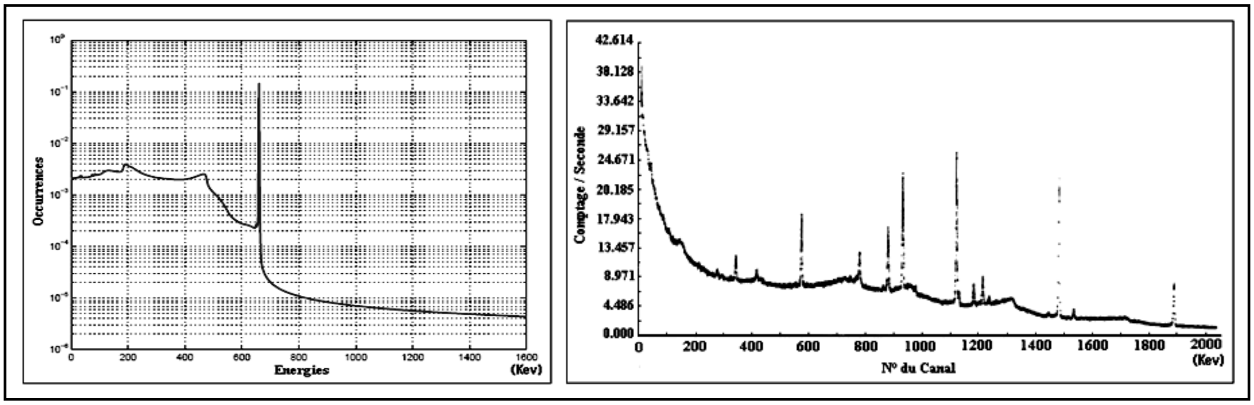


Figure I.3 Simulated and real time gamma ray energy spectrum of Cesium 123 [6].

We notice the appearance of other parasitic lines as well as an enlargement tangible features of the main line within the real spectrum. These measurement anomalies are due to the hidden presence of several types of disturbances which degrade the ideal line spectrum such as: perturbations resulting from the interaction of incident photons with the material of the detector, disturbances from the instrumentation chain and signal acquisition, and disturbances resulting from the random nature of the incident signal…etc. [6].

I.5 Spectroscopy system architecture

The principle of a nuclear Gamma ray detection chain is to convert each photon, traversing the sensitive zone of the detector, into a measurable signal, proportional to the energy of the incident photon; therefore, two architectures of spectroscopy system are available in the market [7].

I.5.1 Analog spectroscopy system

In the early stages of nuclear instrument development, the analog spectroscopy system was proposed with the support of analog pulse processing (APP) technology, which mainly includes a germanium (Ge) detector with its cryostat cooled mechanically or by liquid nitrogen, a detector bias source, a preamplifier, signal conditioning circuit, low-speed ADC and processing unit as shown in Fig I.4.

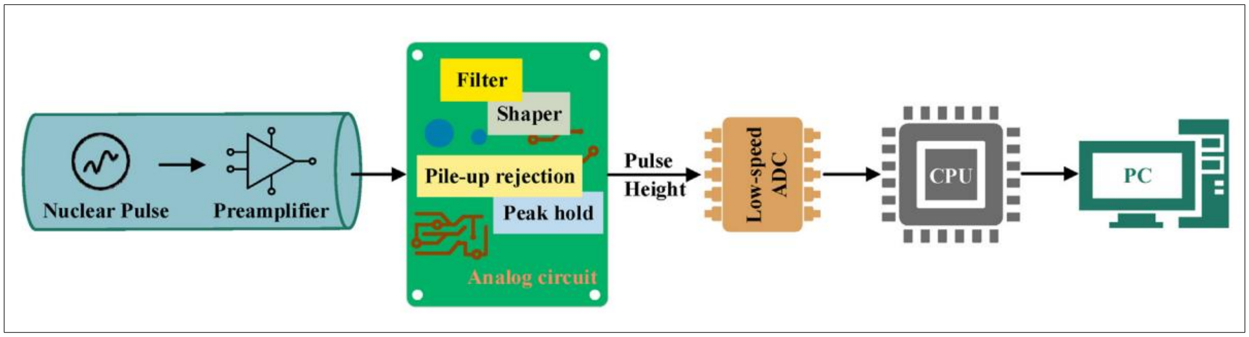


Figure I.4 Architecture of analog spectroscopy system [7].

I.5.2 Digital spectroscopy system

With the development of digital pulse processing (DPP) technology, the digital spectroscopy system has transformed the design of analog circuit into the study of digital algorithm that can be performed by a programmable logic device as shown in Fig 1.5. It basically consists of a germanium (Ge) detector with its cryostat cooled mechanically or by liquid nitrogen, a detector bias source, a preamplifier, high-speed ADC a programmable logic device, and processing unit. Digital spectroscopy system been widely used today for its outstanding accuracy and stability performance. In this project, we are going to work only on digital spectroscopy system.

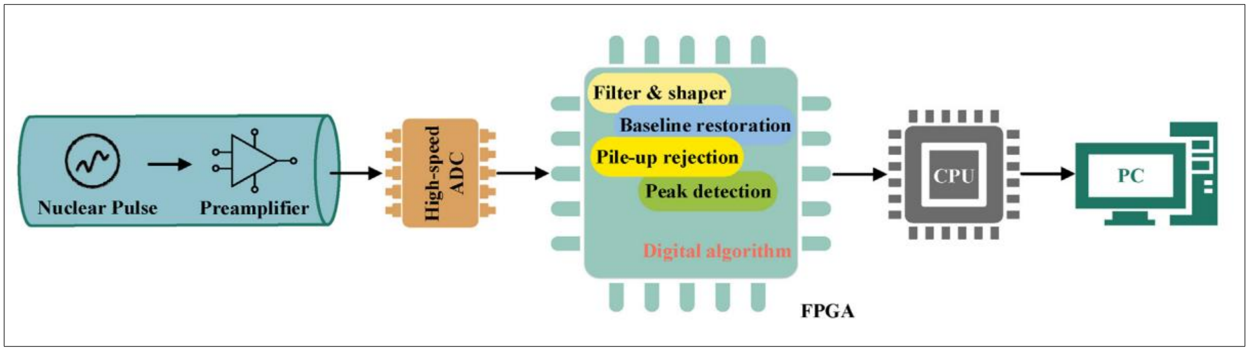


Figure I.5 Architecture of digital spectroscopy system [7].

I.6 Spectroscopic measurement instrumentation

A spectroscopy system consists of a number of stages in cascade that perform specific tasks in order to display the spectrum of the gamma ray source. Generally, a nuclear spectrometer, regardless of the modes and technologies used, performs the following operations:

- Detect the gamma photons and produce a pulse whose height is proportional to the energy deposited in the detector by the gamma ray.
- Linear filtering that minimizes noise of an electrical nature.
- A detection of the maxima of the shaped temporal signal (if the shape of the pulses is constant, the maximum of a pulse is proportional to its integral).
- A classification of the values of these maxima in form of a histogram by a multichannel analyzer (MCA).

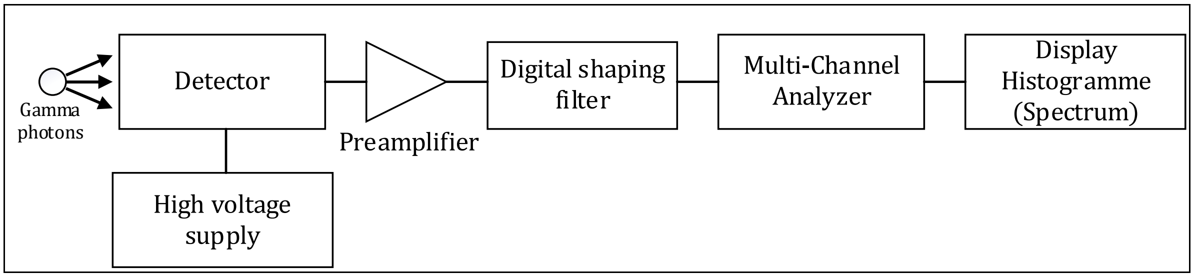


Figure I.6 Simplified block diagram of a digital spectroscopy system.

I.7 High-purity germanium detectors

Detecting the rays consist of generating an electrical signal at the output of the detector that have some characteristics that represent the energy of the corresponding detected photon; however, the radiation cannot be directly measured and it is done using an ionizing radiation sensitive material. And since the interaction of photons with matter depends on the nature of the electromagnetic radiation, each radiation range must be efficiently detected using specific material.

Radiation detectors have two key principles: ionization and excitation [8]:

- In ionization-based detectors, electron-ion pairs are generated by enough energy when ionizing radiation reaches atoms of a sensitive material and removes orbital electrons.
- In excitation-based detectors, bounded electrons are raised to an excited state in the atom or molecule when part of the radiation energy is transferred to them. The electron excited to these states returns to its ground state emitting light in the UV-Visible region.

Many detectors were developed to detect electromagnetic radiations, and they vary in the type of electromagnetic radiation they detect, in the used material type and form and the shape of the output signal…etc., they also differ in resolution, performances, price, size…etc. As shown in

Fig 1.7 the spectrum gamma-ray is different from detector to another. In this thesis we will consider using the **High-purity germanium (HPGe) detectors** and it is the best solution for precise gamma-ray spectroscopy.

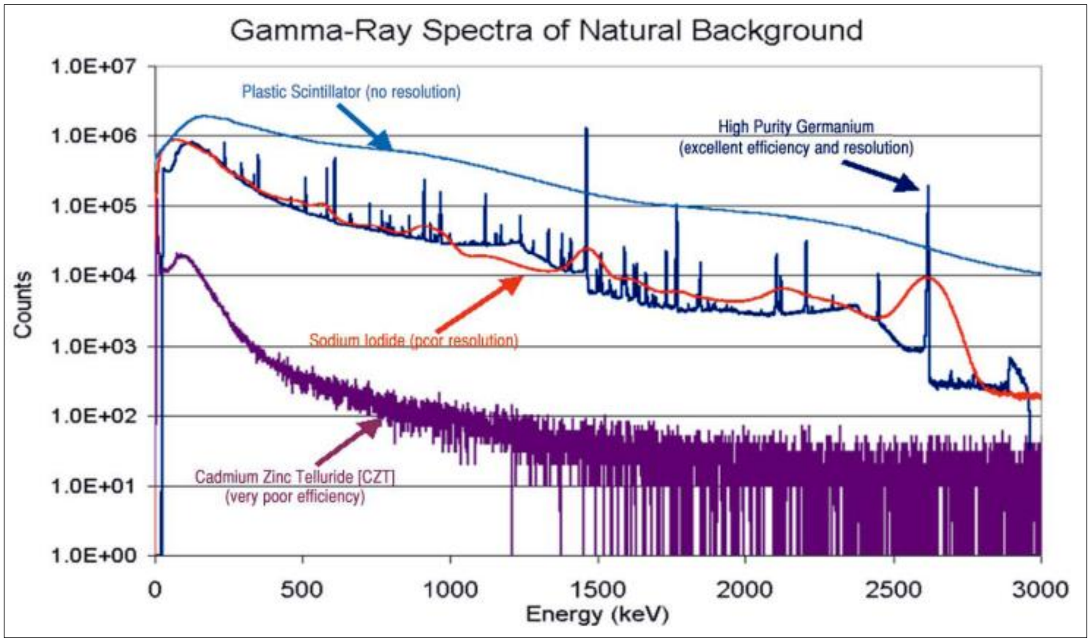


Figure I.7 Gamma ray energy spectrum using different gamma ray detector [9].

HPGe detectors are made by highly refining the element germanium and growing it into a crystal. The crystal goes through a series of processing steps culminating in the attachment of positive and negative contacts, which turns it into a semiconductor diode with a P−I−N structure similar to a standard p−n diode but with an intrinsic region between the p and n electrodes. This intrinsic region is sensitive to ionizing radiation.

The key process by which a gamma ray is detected is ionization, where it gives a part or all of it energy to an electron, therefore, when photons interact with the material within the depleted region, under a reverse bias, electrons are raised to the conduction band leaving behind holes in the valence band producing a large number of electron-hole pairs. As a high voltage is applied across the semiconductor, these charge carriers are readily attracted to the electrodes and moved by the electric field towards the p and n electrodes, which result into a current pulse that flows into the circuit making a current pulse. The height of this pulse is directly proportional to the number of carriers collected, which is proportional to the energy deposited in the material of the detector by the incident photon. The relationship between energy and current is so precise in HPGe detectors that energies are determined to better than 1/10th of 1 percent.

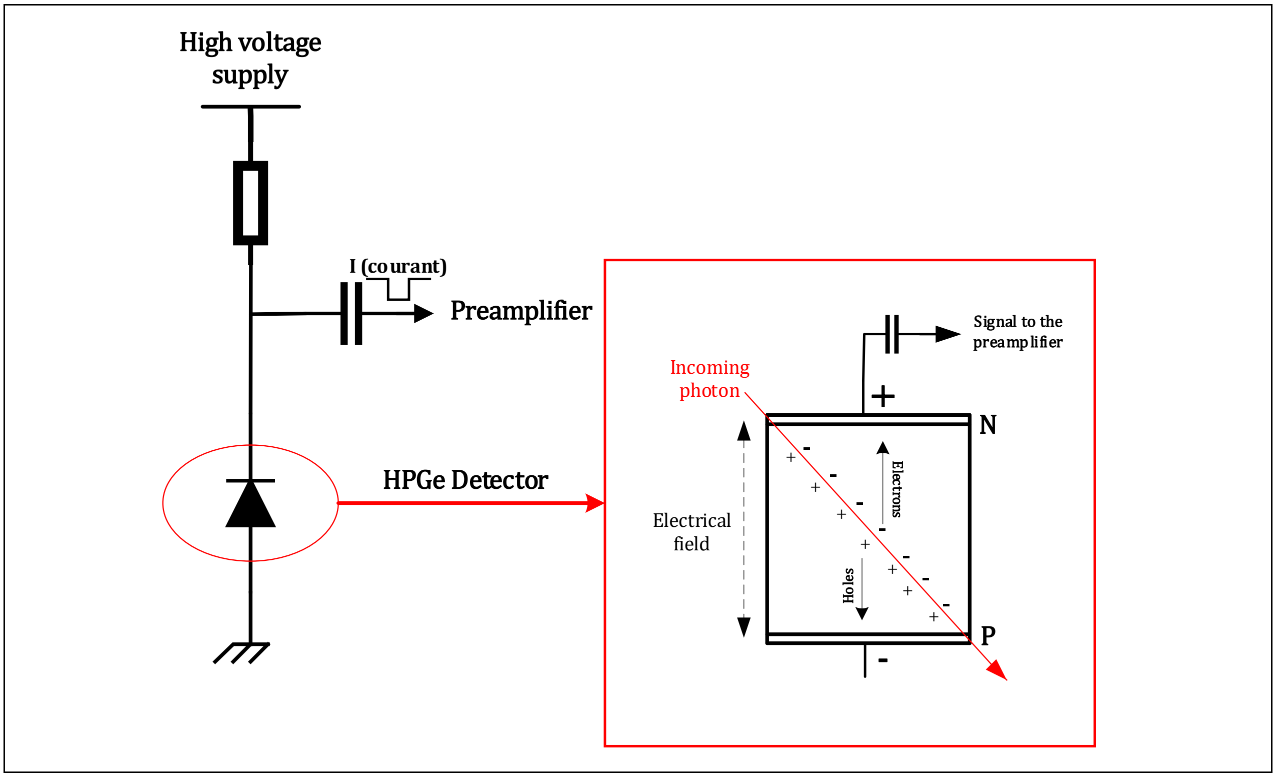


Figure I.8 Simplified scheme of HPGe detector.

Germanium is preferred over silicon because of its higher atomic number, which increases the probability of gamma photons interaction. It also has a relatively low bandgap (less than silicon) which provides better resolution; However, at higher temperatures, electrons can be thermally excited from the valence band to the conduction band creating a thermally generated charge carrier. This results in a leakage current that reduces the resolution of the detector. Therefore, the detectors must be cooled to low temperatures to reduce the thermal activity and reducing the noise by reducing the number of electron-hole pairs in the crystal in the absence of radiation. This cooling is achieved mostly by using liquid nitrogen, which has a temperature of 77K [8]–[10]. For further details of common HPGe configurations, consult [9].

I.8 The Preamplifier

The detector output is a low-amplitude with short-duration current pulse which make it hard to process and require us to use a preamplifier to convert it to a sufficiently long voltage pulse with amplitude proportional to the energy deposited in the detector during the gamma-ray interaction.

The preamplifier is the first component after the detector in a signal processing chain of a radiation detecting. The charge created within the detector is collected by the preamplifier. Despite its name, the preamplifier does not act as an amplifier (just means “before”, i.e., “pre” the amplifier), but acts as an interface between the detector and the pulse processing electronics that follows. The main function of a preamplifier is to extract the signal from the detector without significantly degrading the intrinsic signal-to-noise ratio. Therefore, the preamplifier is located as close as possible to the detector, and the input circuits are designed to match the characteristics of the detector. Two important requirements of the preamp are [11]:

- To terminate the capacitance quickly to maximize the signal-to-noise ratio. The cable length between the preamp and the detector is maintained at minimum as well due to the same reason.
- To have a low output impedance: i.e., to provide a low impedance source for the amplifier. Of course, it should provide a high impedance load for the detector.

Most manufacturers offer several preamplifier models that are optimized for different detector types. Parameters such as noise level, sensitivity, risetime, and count-rate capability may be different for different models [11]. Three basic types of preamplifiers are available: the current-sensitive preamplifier, the parasitic-capacitance preamplifier, and the charge-sensitive preamplifier (CSP); however, we will only focus on the charge-sensitive preamplifiers as they are preferred for most energy spectroscopy applications due to their design that offers low noise, stability, and provides an output proportional to the total charge flowing from the detector during the pulse event. The basic stripped down schematic diagram of an RC feedback charge sensitive preamplifier is shown in Fig I.9.

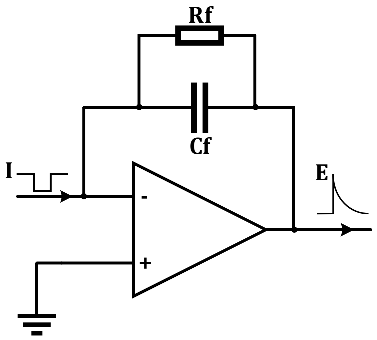


Figure I.9 Simplified scheme of charge sensitive preamplifier

The detector high voltage bias is fed through the preamp in general. When a preamp is ac-coupled, a single cable is connected between the preamplifier and the detector, and is used for both high voltage bias to the detector and signal extraction from the detector. A coupling capacitor should be provided between the detector and preamp circuits in this configuration while it is eliminated if the dc-coupled configuration is adopted instead [11].

In the charge sensitive preamp, the feedback capacitor $\mathrm{C_f}$ between the input and output collect the charges from the detector over a period of time, effectively integrating the detector current pulse. In case of the elimination of the resistance $\mathrm{R_f}$, a step change is produced in the output voltage at each input current pulse forming a staircase pattern with each upward step corresponding to a separate pulse until it reaches its maximum output ; however, the high valued $\mathrm{R_f}$ bleed resistor (feedback resistor) that is parallel with the feedback capacitor works as a reset for the charge sensitive preamp and it transforms upward step response caused by capacitor $\mathrm{C_f}$ to a tail pulse at the output of the preamp that have a rise time equal to the detector current pulse width ($\sim 100\mathrm{ns}$) in the ideal case (τ_R is the rise time constant) and a slow decay time constant $\tau_\mathrm{d} = \mathrm{R_f C_f}$ ($\sim 100\mu\mathrm{s}$) and a peak value $\mathrm{E} = -\frac{\mathrm{Q_D}}{\mathrm{C_f}}$ where :

$$\mathrm{Q_D} = \frac{\mathrm{E_\gamma . e. 10^6}}{\varepsilon} \qquad \text{I.6}$$

$\mathrm{E_\gamma}$ is the energy in MeV of the incident radiation, e is the charge of an electron (-1.6×10^{-19} coulomb), 10^6 converts MeV to eV, and ε is the amount of energy required to produce an electron-hole pair in the detector. Approximate values of ε for the HPGe detector is $\varepsilon = 2.96$ at the temperature of 77K [12]. As can be seen the output pulse height is in proportion to the energy deposited by the radiation interaction. Accordingly, by calculating the peak value we can easily determine the energy of the incident gamma photon.

The form of the preamplifier output can be written as:

$$V_{preamp}(t) = E\left(e^{-\frac{t}{\tau_d}} - e^{-\frac{t}{\tau_R}}\right) \qquad \text{I.7}$$

It also should be pointed out that the polarity of the output pulse is positive when the current pulse flows from the CSP input to the HPGe output and negative when the detector current flows into the CSP. The feedback resistor R_f has an intrinsic noise (Johnson noise) associated with it. The noise can be minimized by selecting a higher R_f value, which is limited because may lead to too long a time constant [11].

Note that the schematic in the Fig 1.9 of the CSP shows only the basic parts of it and in a commercial CSP, an output amplifying stage can be added after the main stage. Also, it can include an internal pole–zero cancellation circuit.

I.9 Sources of error in a spectroscopic measurement chain

There are a variety of sources of errors in the spectroscopic chain, for that we will take an overview of the main sources of errors in the system and later on the digital shaping filter is used to correct or at least to minimize these errors.

I.9.1 Electrical noises

The spectrophotometric chain is subject to different electrical noises that came from different sources making them hard to be quantified, these noises came from external sources such as atmospheric (atmospheric irregularities, extraterrestrial or industrial noise, or internal such as:

- Noise from the input capacitance caused by cabling, ground loops, microphonics or radio-frequency pickup.
- The thermal noise also called Johnson noise that comes from the fluctuations affecting the trajectories of the charge carriers in the semiconductors due to the interactions.
- Shot noise (or Schottky noise) where the current is not considered as a uniform flux, but as the composition of a large number of elementary pulses.
- Lorentzian noise
- Flicker noise which is associated with the presence of direct (analog) voltage or direct current passing through the component and it depends on the frequency.
- And other noise sources that can be neglected in the spectroscopy system.

I.9.2 Ballistic deficit effect

In addition to the noise there are other phenomena that reduce the energy resolution of a spectroscopy system such as the ballistic deficit effect.

As explain before, the CSP capacitor C_f collect the charges from the detector over time during the pulse event, and discharge through the bleed resistor $\mathrm{R_f}$. Based on this simple description we can conclude that during the collection charge phase a part of the detector charges will be dissipated through$R_f$leading to a reduction in the recorded pulse height in the output of the preamplifier.

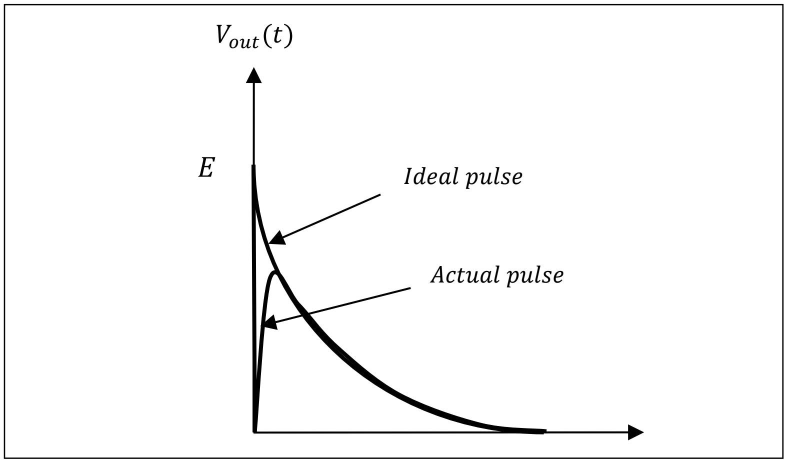


Figure I.10 Pulse-height loss due to ballistic deficit effect.

Therefore, the preamplifier output pulse height is related to the charge collected for the integration time of the amplifier and the Ballistic deficit is the term used to describe the reduction in the recorded pulse height due to variations in the charge collection. If the system has ballistic deficit effects, variations in detector charge collection time will produce poor resolution and poor peak shape. Such effects are most common in larger detectors[13]. However, many research was made on that and one of the techniques to solve this pulse-height loss that cause a calculation error of the energy of the incident gamma photon is to use a flat top shaper after the preamplifier to detect the real peak.

I.9.3 Pile up effect

The spectroscopy chain also suffers from the pile-up effect, the exponential decay pulse at the output of the preamplifier of a single detected gamma photon has a short duration and not zero, this means, if two photons are coincidently emitted by the gamma ray source closely enough, the pulses pile on top of each other as new signals arrive before the old ones have decayed and that will make the peak of the piled pulse higher than what is supposed to be and decrease the energy resolution of the spectrometer and in reality the probability of this effect almost exist in all the time.

The pulse pile up events are formed by random coincidence due to photoelectric absorption, Compton scattering, backscatter interactions, and so on. The probability of the pulse pileup depends on the count rate and the dead time of the detector, thus, with a large dead time the probability of the pileup increases. The pileup effect also causes distortion of the amplitude of the pulse, resulting in deterioration of the energy spectrum. Various methods are being studied to solve the pulse pileup problem[14].

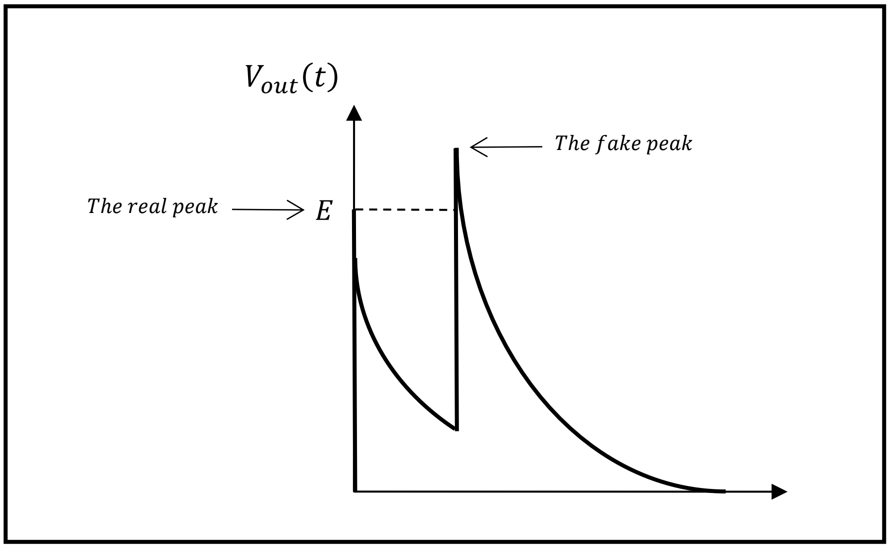
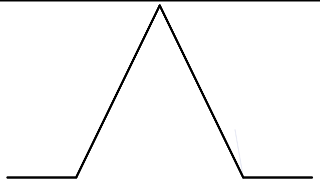
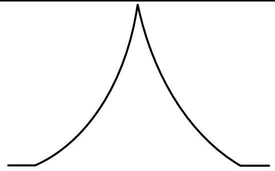
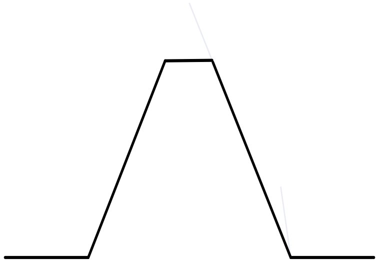
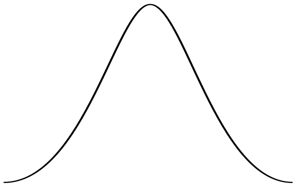
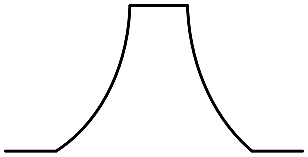


Figure I.11 Pule up pulses

I.10 Shaper filters

As discussed previously, the preamplifier’s exponential output pulse is subject to many undesirable effects and problems such as noises from various sources, ballistic deficit effect and pile-up problem; therefore, analog and digital shapers were developed in order to solve or at least to reduce these problems and gives a better result in measuring the energy, such as: Gaussian, cusp-like, flat-topped cusp-like, trapezoidal and triangular shapes...etc. For our project we will use a digital trapezoidal pulse shaping as it is easy to understand and implement in digital logic and provides a good performance.

Table I.1 Different types of the shaped pulse.

PULSE SHAPE		
 Triangular	 Cusp-like	 Trapezoidal
 Gaussian	 Flat-topped cusp-like	

I.11 Multichannel analyzer

Using the shaping filter allows to correct the peak losses and increase the signal to noise ratio; therefore, by measuring the height of the shaped signal we can detect the energy of the emitted gamma photon. The Multichannel analyzer allows classifying the detected peaks of each detected gamma photon according to their individual energy in order to display the spectrum which is a histogram that represent the numbers of gamma photons that belongs to each channel where each channel represent a specific level of energy.

I.12 Conclusion

In this chapter, we gave an overview about some of the physical phenomenon in the spectrum of gamma ray. We also explained the task of each stage in a digital spectroscopy system and the variety of problems that occur and its sources in the spectroscopy measurement chain. The errors and noises in the spectroscopic chain come from different sources and effect the resolution of the measurement which forces the use of digital shaping filters to solve or minimize those problems.

Chapter II

Digital Trapezoidal Filter and Digital Processing of The Pulse

II.1 Introduction

Using the preamplifier exponential output pulse directly to measure the energy decrease the resolution of the spectrum which require the use of digital shapers to increase it. In this chapter, we will discuss one of the most efficient digital shapers known as “Trapezoidal filter” by deriving its transfer function that allows a deeper understanding of the filter and the choice of the optimal parameters. In the existence of a gamma ray source, many gamma photons will be detected which will result to a series of pulses that we will also discuss how to detect their peaks and classify them.

II.2 Digital trapezoidal filter

Trapezoidal filtering is well known and efficient digital shaper that converts a digitized exponential pulse into a trapezoidal pulse with adjusted parameters $A$and B as Fig II.1 shows.

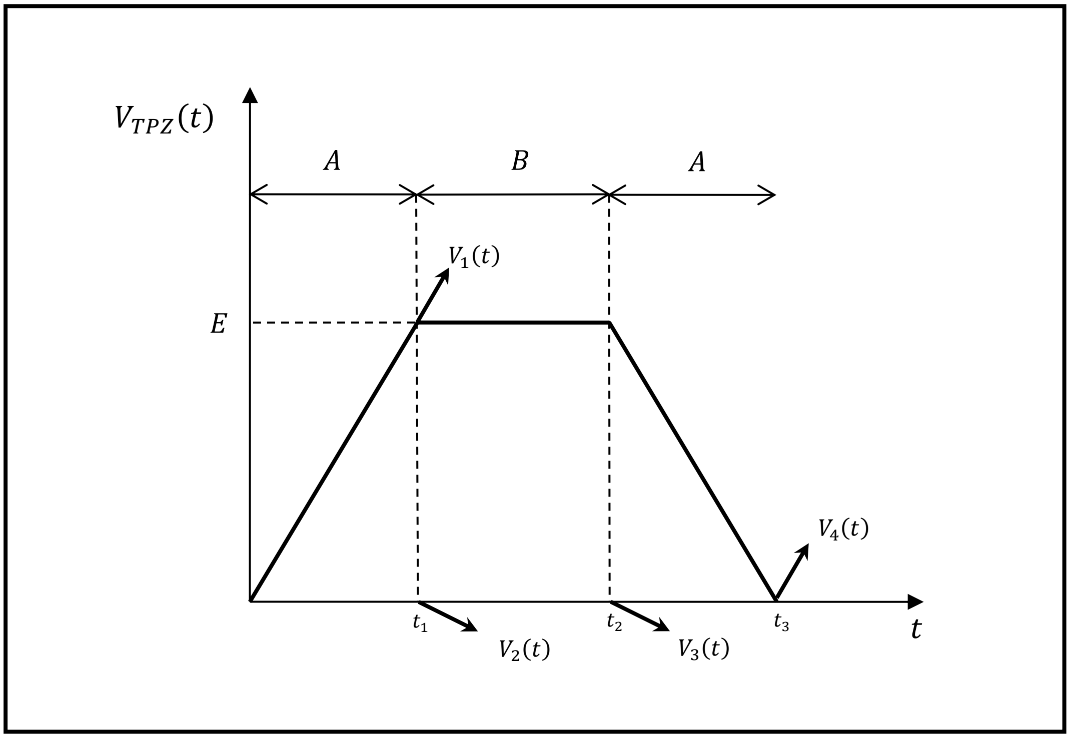


Figure II.1 The ideal trapezoidal output of the DTF.

The digital trapezoidal shaping (DTS) filter has the advantages in ballistic deficit correction, energy resolution by improving the signal-to-noise ratio (SNR), pulse throughput by allowing the pile-up pulse separation, which is often used to process the digital nuclear signal in the gamma-ray spectroscopy systems.

Fig II.2 represents how the trapezoidal filter can correct the pulse height loss that is due to ballistic deficit. By choosing the right parameters for the DTS filter we can ensure that the end of the flat top will reach the peaking value of the real pulse.

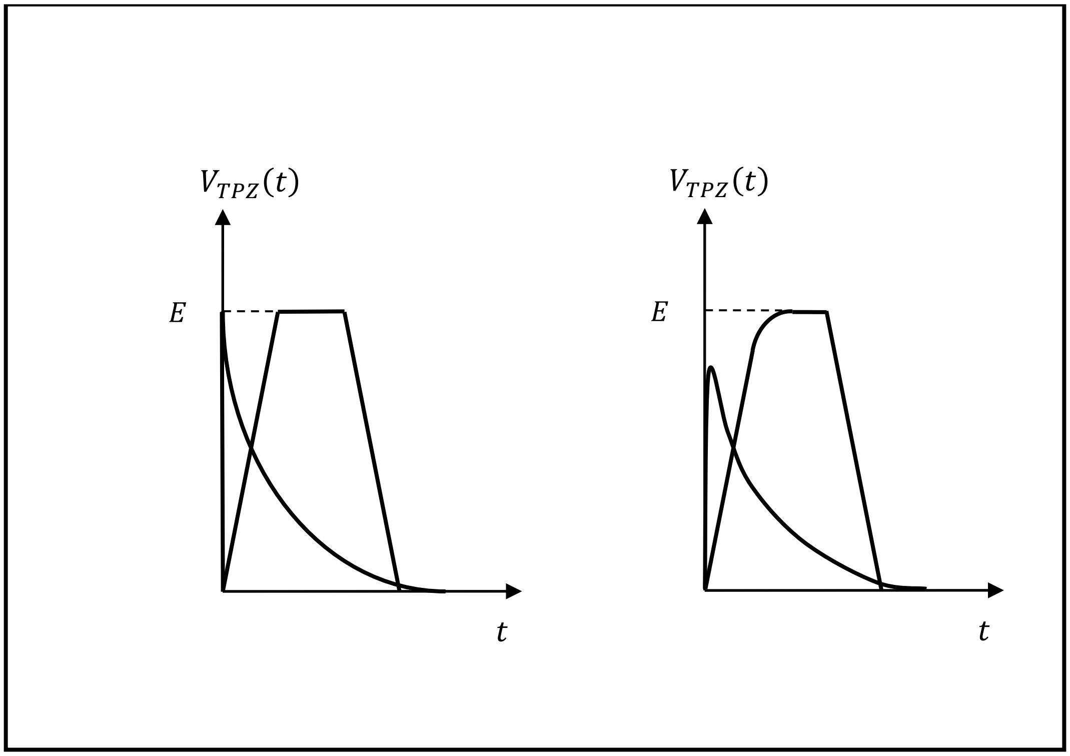


Figure II.2 The effect of the ballistic deficit in the trapezoidal shaped pulse.

Fig II.3 represents how the trapezoidal filter can separate the pile-up pulses if the choice of DTS parameters is suitable to the trapezoidal output finish before the pilled-up pulse occurs.

Many algorithms are developed through researches to perform the DTS as represented in [15]–[19]. The one that we will select as our DTS filter is based on developing the discrete transfer function of the trapezoidal filter as it is easy to understand and to analyze.

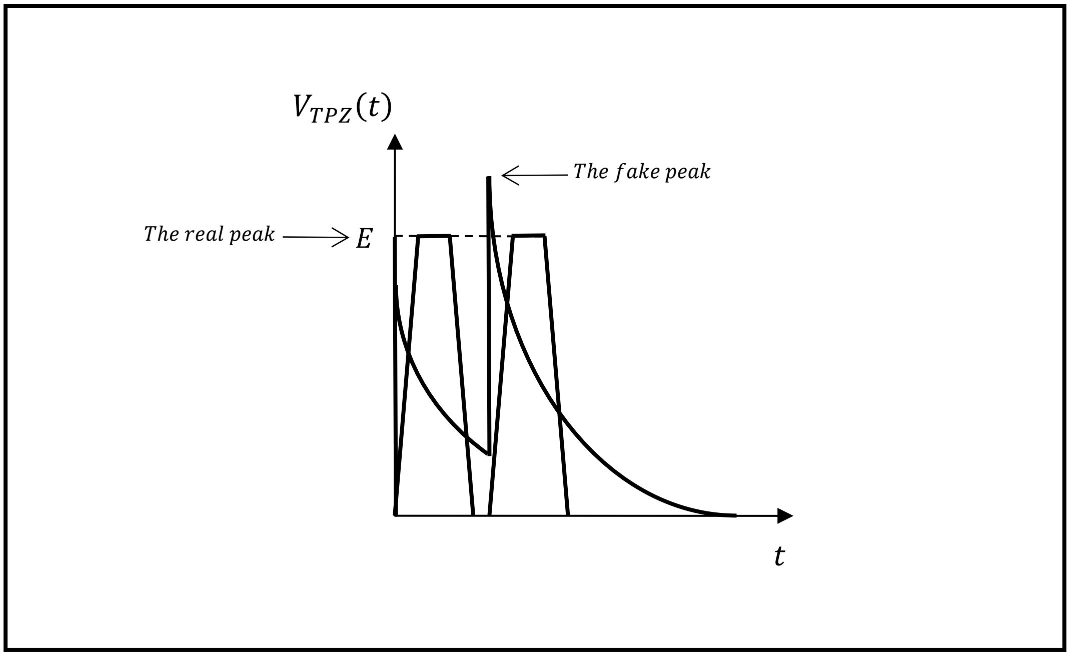


Figure II.3 The advantage of trapezoidal shaped pulse in eliminating the pile up pulses.

II.3 Developing the discrete time TF and the recursive equations of the DTS filter

The algorithm can be obtained by Z transform method which is easy to derive. The function of the DTS filter is to take a discrete exponential input pulse signal V_{IN} from the ADC and convert it to a trapezoidal shaped output V_{TPZ}.The form of the exponential input pulse signal V_{in} from the preamplifier can be writen by the following time domain function:

$$V_{in}(t) = V_{preamp}(t) = E\left(e^{-\frac{t}{\tau_d}} - e^{-\frac{t}{\tau_R}}\right) \qquad \text{II.1}$$

τ_d is the decay time constant, τ_R Is the rise time constant of the preamplifier pulse and $E$is the ideal amplitude of the pulse neglecting the rise time leads to:

$$V_{in}(t) = E\left(e^{-\frac{t}{\tau_d}}\right) \qquad \text{II.2}$$

By discretizing the preamplifier pulse by applying Z transform to the simplified form of $V_{in}(t)$ we get:

$$V_{IN}(z) = E\left(\frac{z}{z-\beta}\right) = E\left(\frac{1}{1-\beta z^{-1}}\right) \qquad \text{II.3}$$

$\beta = e^{-\frac{\Delta t}{\tau_d}}$ and Δt is the simpling time.

As can be seen in the Fig II.1, a trapezoid wave may be synthesized by the sum of four terms:

$$V_{TPZ}(t) = \sum_{i=1}^{4} V_i(t) = V_1(t) + V_2(t) + V_3(t) + V_4(t) \qquad \text{II.4}$$

$$V_{TPZ}(t) = V_1(t) - V_1(t - t_1) - V_1(t - t_2) + V_1(t - t_3) \qquad \text{II.5}$$

The linear function $V_1(t)$ is expressed in the time domain and Z-transform domain as following:

$$V_1(t) = \frac{E}{t_1} t \Rightarrow V_1(z) = \left(\frac{E}{A}\right) \frac{z}{(z-1)^2} \qquad \text{II.5}$$

By assuming symmetrical shape of the trapezoid, the following time assignments are given:

$$t_1 = A\Delta t\,, t_2 = (A+B)\Delta t\,, t_3 = (2A+B)\Delta t \qquad \text{II.6}$$

Where A is the trapezoid edges, duration expressed in the number of sampling time Δt.and B is the trapezoid flat top duration expressed in the number of sampling time Δt. The remaining three linear pulses can be expressed by means of a delayed $V_1(t)$ function with adequate polarities

$$V_2(t) = -V_1(t - t_1) \rightarrow V_2(z) = -V_1(z) z^{-A} \qquad \text{II.7}$$

$$V_3(t) = -V_1(t - t_2) \Rightarrow V_3(z) = -V_1(z) z^{-(A+B)} \qquad \text{II.8}$$

$$V_4(t) = +V_1(t - t_3) \Rightarrow V_4(z) = +V_1(z) z^{-(2A+B)} \qquad \text{II.9}$$

By applying Z-transform to equation II.4 and replacing the terms II.7, II.8 and II.9 in it we get:

$$V_{TPZ}(z) = V_1(z) + V_2(z) + V_3(z) + V_4(z) \qquad \text{II.10}$$

$$V_{TPZ}(z) = V_1(z) - V_1(z) z^{-A} - V_1(z) z^{-(A+B)} + V_1(z) z^{-(2A+B)} \qquad \text{II.11}$$

$$V_{TPZ}(z) = \left(\frac{E}{A}\right) (1 - z^{-A} - z^{-(A+B)} + z^{-(2A+B)}) \frac{z}{(z-1)^2} \qquad \text{II.12}$$

$$V_{TPZ}(z) = \left(\frac{E}{A}\right)(1 - z^{-A})\left(1 - z^{-(A+B)}\right)\frac{z^{-1}}{(1 - z^{-1})^2} \quad \text{II.13}$$

Therefore, the discrete transfer function of the filter is given as following:

$$H_{TPZ}(z) = \frac{V_{TPZ}(z)}{V_{IN}(z)} \quad \text{II.14}$$

$$H_{TPZ}(z) = \frac{\left(\frac{E}{A}\right)(1 - z^{-A})\left(1 - z^{-(A+B)}\right)\frac{z^{-1}}{(1-z^{-1})^2}}{E\left(\frac{1}{1-\beta z^{-1}}\right)} \quad \text{II.15}$$

$$H_{TPZ}(z) = (1 - \beta z^{-1})\frac{(1 - z^{-A})}{(1 - z^{-1})}\frac{\left(1 - z^{-(A+B)}\right)}{(1 - z^{-1})}\left(\frac{z^{-1}}{A}\right) \quad \text{II.16}$$

$$H_{TPZ}(z) = \frac{H_1(z)H_2(z)H_3(z)H_4(z)}{H_5(z)H_6(z)} \quad \text{II.17}$$

Where:

$$H_1(z) = (1 - \beta z^{-1}), H_2(z) = (1 - z^{-A}), H_3(z) = \left(1 - z^{-(A+B)}\right) \quad \text{II.18}$$

$$H_4(z) = \left(\frac{z^{-1}}{A}\right), H_5(z) = (1 - z^{-1}), H_6(z) = (1 - z^{-1}) \quad \text{II.19}$$

Writing the discrete transfer function in this form allows us to split the filter into four cascaded stages as illustrated in Fig II.4 that are easy to analyze and to understand as presented in Table II.1.

$$H_{TPZ}(z) = H_{s1}(z)H_{s2}(z)H_{s3}(z)H_{s4}(z) \quad \text{II.20}$$

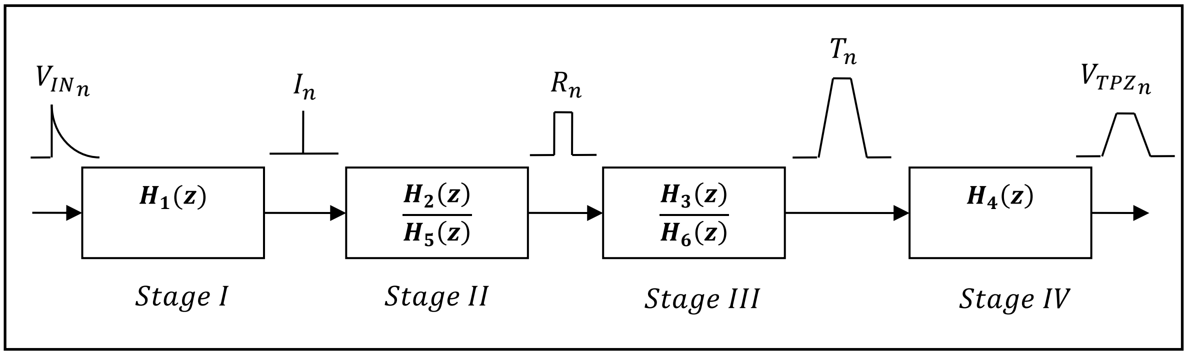


Figure II.4 The different processing stages of the digital trapezoidal shaping filter.

II.4 Deriving the recursive formula of each stage

II.4.1 Stage 1

$$\frac{I_n}{V_{IN_n}} = (1 - \beta z^{-1}) \Rightarrow I_n = V_{IN_n}(1 - \beta z^{-1}) \qquad \text{II.21}$$

$$I_n = V_{IN_n}(1 - \beta z^{-1}) \Rightarrow I_n = V_{IN_n} - \beta z^{-1} V_{IN_n} \qquad \text{II.22}$$

$$I_n = V_{IN_n} - \beta z^{-1} V_{IN_n} \Rightarrow I_n = V_{IN_n} - \beta V_{IN_{n-1}} \qquad \text{II.23}$$

II.4.2 Stage 2

$$\frac{R_n}{I_n} = \frac{(1 - z^{-A})}{(1 - z^{-1})} \Rightarrow R_n(1 - z^{-1}) = I_n(1 - z^{-A}) \qquad \text{II.24}$$

$$R_n(1 - z^{-1}) = I_n(1 - z^{-A}) \Rightarrow R_n - z^{-1}R_n = I_n - z^{-A}I_n \qquad \text{II.25}$$

$$R_n - z^{-1}R_n = I_n - z^{-A}I_n \Rightarrow R_n = z^{-1}R_n + I_n - z^{-A}I_n \qquad \text{II.26}$$

$$R_n = z^{-1}R_n + I_n - z^{-A}I_n \Rightarrow R_n = R_{n-1} + I_n - \ I_{n-A} \qquad \text{II.27}$$

II.4.3 Stage 3

$$\frac{T_n}{R_n} = \frac{\left(1 - z^{-(A+B)}\right)}{(1 - z^{-1})} \Rightarrow T_n(1 - z^{-1}) = R_n\left(1 - z^{-(A+B)}\right) \qquad \text{II.28}$$

$$T_n(1 - z^{-1}) = R_n\left(1 - z^{-(A+B)}\right) \Rightarrow T_n - z^{-1}T_n = R_n - z^{-(A+B)}R_n \qquad \text{II.29}$$

$$T_n - z^{-1}T_n = R_n - z^{-(A+B)}R_n \Rightarrow T_n = z^{-1}T_n + R_n - z^{-(A+B)}R_n \qquad \text{II.30}$$

$$T_n = z^{-1}T_n + R_n - z^{-(A+B)}R_n \Rightarrow T_n = T_{n-1} + R_n - \ R_{n-(A+B)} \qquad \text{II.31}$$

II.4.4 Stage 4

$$\frac{V_{TPZ}}{T_n} = \left(\frac{z^{-1}}{A}\right) \Rightarrow A.V_{TPZ} = z^{-1}T_n \qquad \text{II.32}$$

$$A.V_{TPZ} = z^{-1}T_n \Rightarrow V_{TPZ_n} = \frac{1}{A}T_{n-1} \qquad \text{II.33}$$

Table II.1 Representation of each stage

Filter stage	Factor in Transfer function	Recursive relation	Signal shape at Output of the stage
1	$(1-\beta z^{-1})$	$I_n = V_{IN_n} - \beta V_{IN_{n-1}}$	Impulse
2	$\frac{(1-z^{-A})}{(1-z^{-1})}$	$R_n = R_{n-1} + I_n - \ I_{n-A}$	Rectangular uni-polar
3	$\frac{\left(1-z^{-(A+B)}\right)}{(1-z^{-1})}$	$T_n = T_{n-1} + R_n - \ R_{n-(A+B)}$	Trapezoidal
4	$\left(\frac{z^{-1}}{A}\right)$	$V_{TPZ_n} = \frac{1}{A} V_{TPZ_{n-1}}$	Normalized trapezoidal

II.5 Optimal choice of trapezoidal shaping parameters

There are three parameters in the DTS algorithm:A, B and β. A represents the rise time of the trapezoidal pulse and B represent the flat top duration. And as the rise time and fall time are equal, the trapezoidal pulse width is $2A + B$. The parameter β depends only on the sampling time Δt and the characteristics of the preamplifier and can be calculated simply by the following formula: $\beta = e^{-\frac{\Delta t}{\tau_d}}$

The optimal choice of A and B have to consider the noise filtering, ballistic deficit correction, pulse pile-up elimination and amplitude extraction, in order to guarantee the maximal throughput and the best energy resolution in the digital nuclear spectrometer system [20].

Maximum achievable throughput and system resolution are related to the filter width; in general, reducing the filter width increases throughput, but worsens the resolution, mainly due to the series noise contribution to the total noise. The risetime A is adjusted for the minimum noise unless the count rate is high enough to cause significant pileup losses. Shorter risetimes may then be used to achieve higher throughput with somewhat higher noise. If the flattop width B is adjusted to a value longer than the longest detector charge collection time, ballistic deficit effects can be essentially eliminated and it can be adjusted to give the best tradeoff between throughput and resolution [13].

II.6 Restrain the baseline drift

The accumulation of noise can lead to the baseline drift of trapezoidal pulse. Many ways were represented through researches to solve this issue and filtering the original signal before shaping is a way of Restrain the baseline drift. Digital $S-K$ filter performs well in signal processing taking into account amplitude and frequency filtering factors. The true height of filtered signal can be obtained by adjusting amplitude of the filtering factor properly. The algorithm of digital S-K filter is written as [21]:

$$V_{IN_{S-K}}[n] = \begin{cases} \dfrac{(k.(3-a)+2k^2).V_{IN_{S-K}}[n-1]-k^2.V_{IN_{S-K}}[n-2]+a.V_{IN}[n]}{1+k.(3-a)+k^2} & ifn > 0 \\ 0 & otherwise \end{cases} \qquad \text{II.34}$$

Where $V_{IN}[n]$is the discrete input signal, $V_{IN_{S-K}}[n]$ is the output signal, k is the frequency filtering factor and a is the amplitude filtering factor. The original signal $V_{in}(t)$ is filtered by the digital $S-K$ filter, and output of the filter is processed by calling Eq. II.16 recursively. The baseline drift of trapezoidal pulse can be removed by selecting the optimal k and a gains, as shown in Fig II.5.

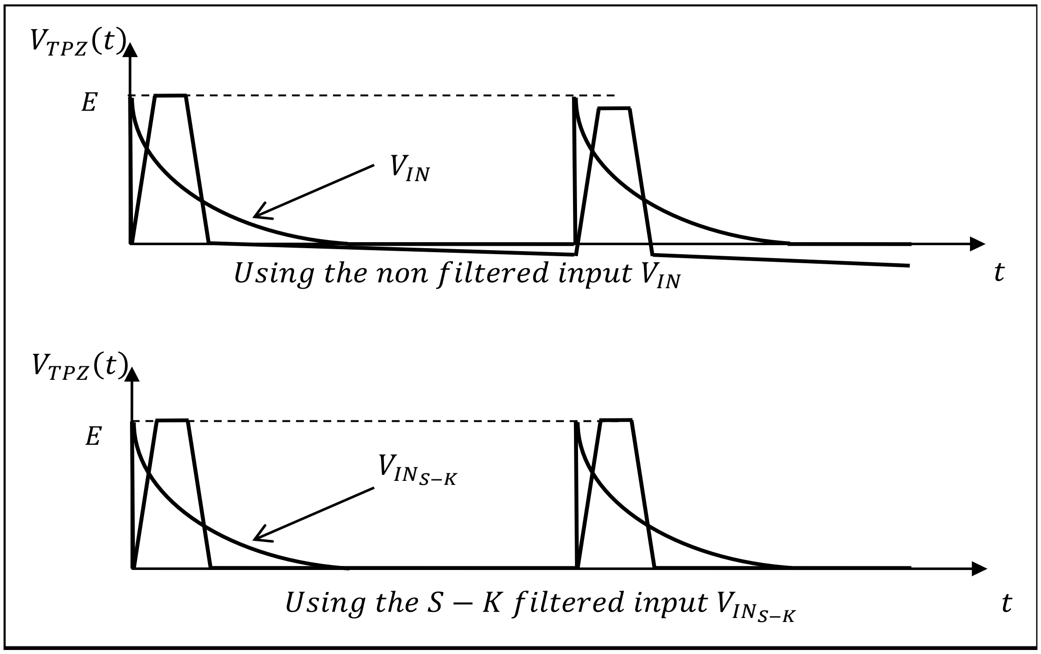


Figure II.5 The base line drift correction of the trapezoidal shaped pulse.

II.7 Measuring the maxima of the signal

Measuring the exponential decay pulse height is performed using the trapezoidal output by averaging a number of samples that are close to the end of the flat top as shown in Fig II.6.

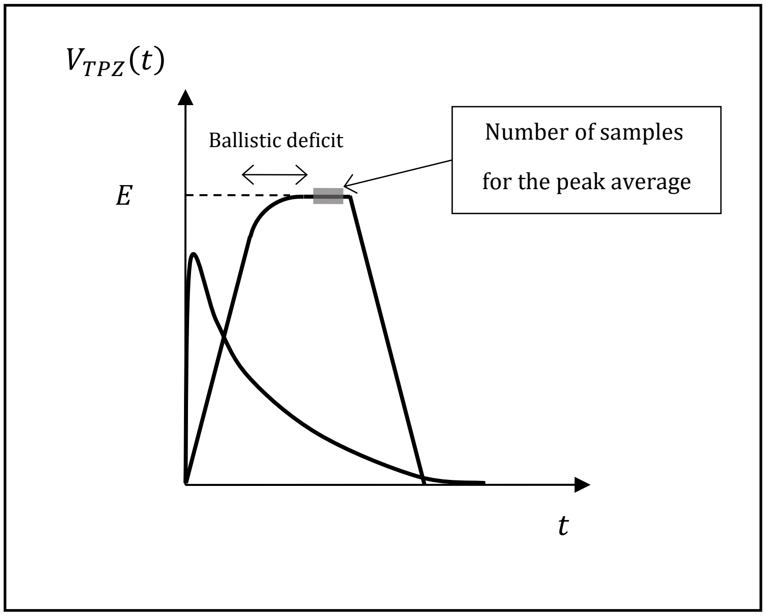


Figure II.6 The peak average at the flat top of the trapezoidal output

To do that first this averaging region must be detected and then the average is calculated. This can be done by using an algorithm that is based on the derivative of $V_{TPZ}(t)$, the Fig II.7 shows the derivative of an ideal trapezoidal output.

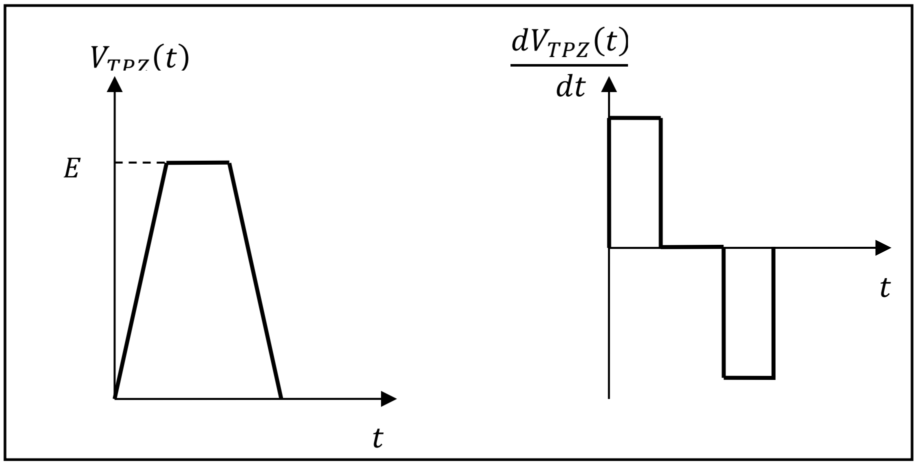


Figure II.7 The derivative of an ideal trapezoidal output.

What the algorithm will do is that whenever it detect a serie of a negative values in the derivative of the trapezoidal filter it start averaging a specific number of previous values of the trapezoidal output, the Fig II.8 shows how that can be implemented in form of blocks:

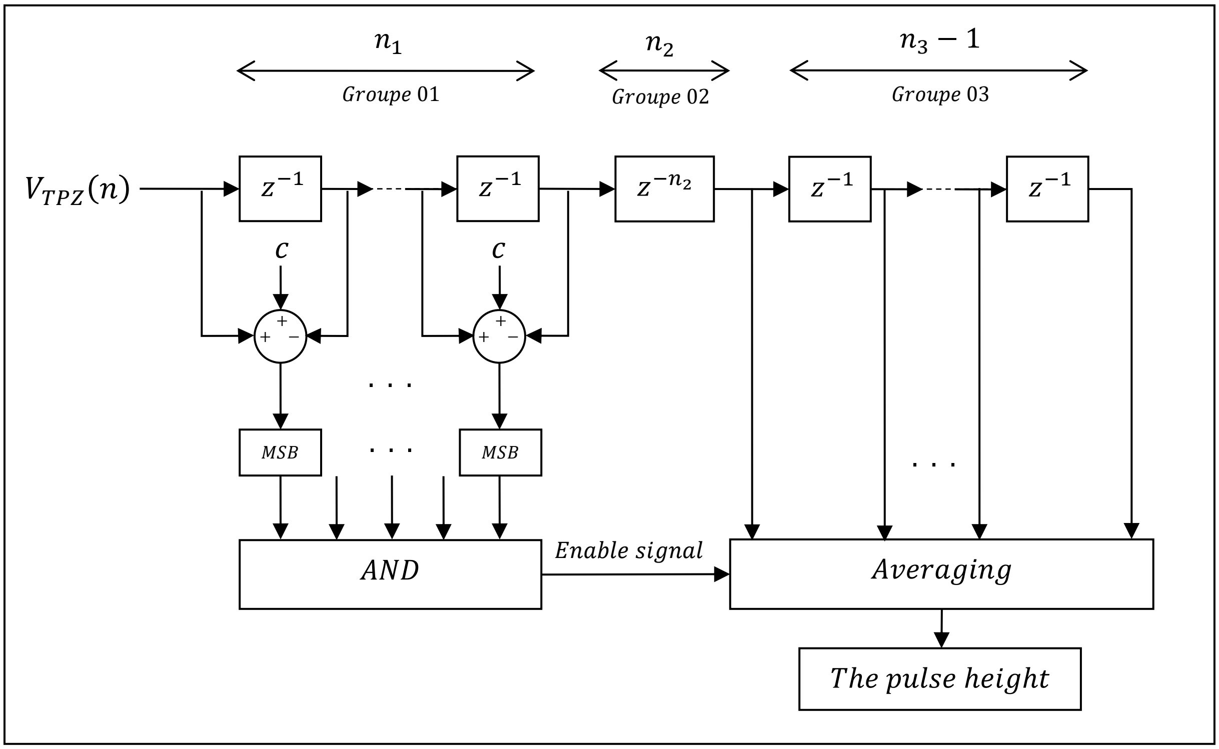


Figure II.8 The block diagram of the height measurement of the trapezoidal shaped pulse.

Where :

n_1: is number of samples to detect the falling of the trapezoidal output.

n_2: is number of samples to not use that are at the end of the flat top.

n_3: is number of samples to be averaged.

c : is a small number.

- The derivative is calculated by $V_{TPZ}(n) - V_{TPZ}(n-1)$.
- c is a small number that is used to eliminate the fake trapezoidal falling detection that can be due to the derivative of the noise.
- The MSB is used to determine the sign of the derivative as it is set to 1 when the sign is negative and 0 when the sign is positive.
- The AND gate have n_1 input and it is used to output a logical when all the n_1 derivative values are negative.

- The output of the AND gate is used to enable the storing of the calculated value of the average that is the pulse height (the peak value).

- The peak is detected only at the rising edge of the AND gate output.

II.8 Base line restoration correction

A DC offset can exist in the baseline of the pulse shaping due to variations in temperature and bias in the instrumentation as well. Therefore, a baseline restorer is recommended to be designed to remove the DC offset to accurately extract the peak location of the pulse signals. The DC offset can be calculated by digital averaging of the trapezoid wave in cases when the spectrometric channel is not busy (zero-area pulse shaping). The accurate pulse height is obtained by subtracting the raw pulse height of the trapezoidal energy filter with the level of average baseline and one of the ways to do that is by using the delaying (shifting the trapezoidal output line) as shown in Fig II.9 [22], [23].

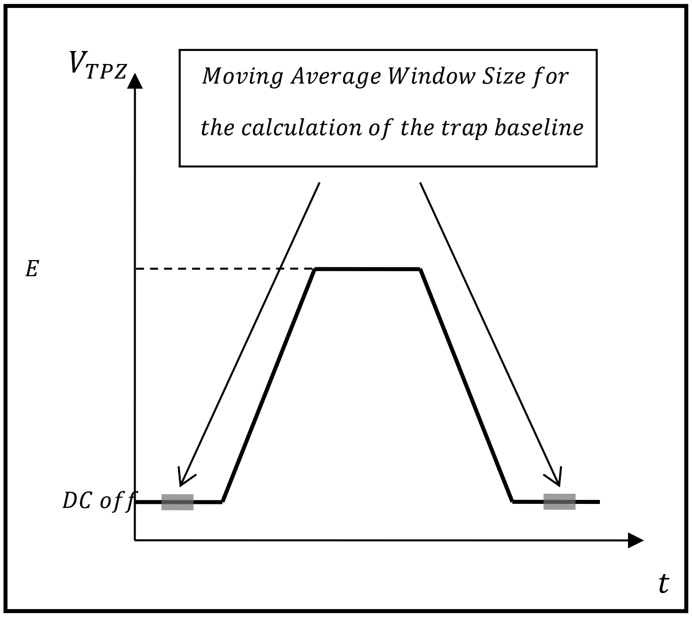

V_{TPZ}
Moving Average Window Size for
the calculation of the trap baseline
E
$DC\ off$
t

Figure II.9 The base line restoration of the trapezoidal shaped pulse using moving average window.

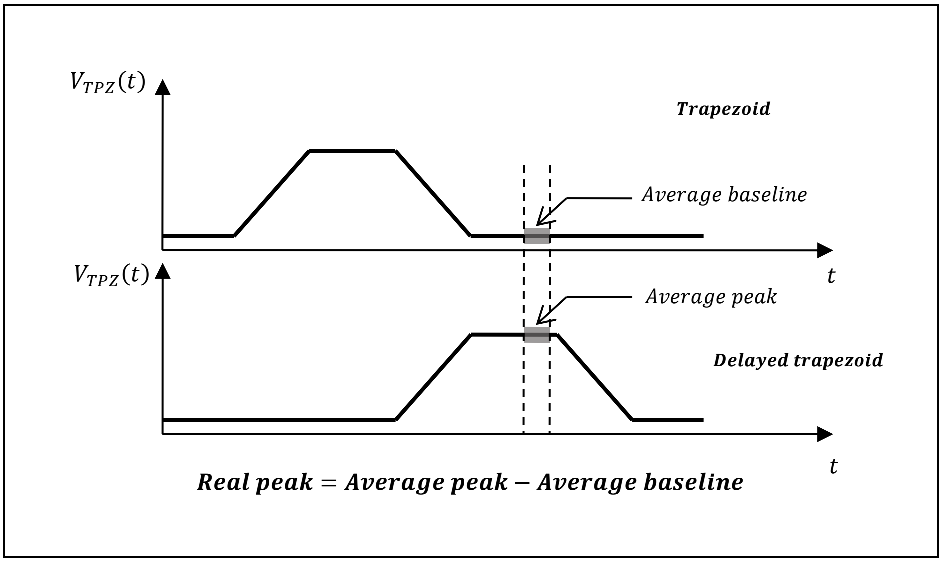


Figure II.10 The base line restoration of the trapezoidal shaped pulse using a delayed pulse.

In case of using the shown technique in Fig II.10 the reading of the peak is delayed but it is synchronized.

It should be pointed out that in this thesis for the simulation and the experimental parts the *S-K* filter and the baseline restoration correction were not used for simplicity reason.

II.9 Classifying the maxima

Many gamma photons with different energy will hit the active area of the HPGe detector resulting a series of exponential decay pulses with different amplitude at the output of the preamplifier this will also result a series of trapezoidal signals where in each one of them the peak value that represent the energy of the detected gamma photon is calculated. The last task is to count how many maxima values belongs to each channel as shown in Fig II.11. The maximum number of channels that can be represented is defined by the bits width of the output of the trapezoidal filter.

After having the number of the maximas that belongs to each channel, a software can be used to plot a histogram that shows the number of counts in function of the channels which is the spectrum in real time.

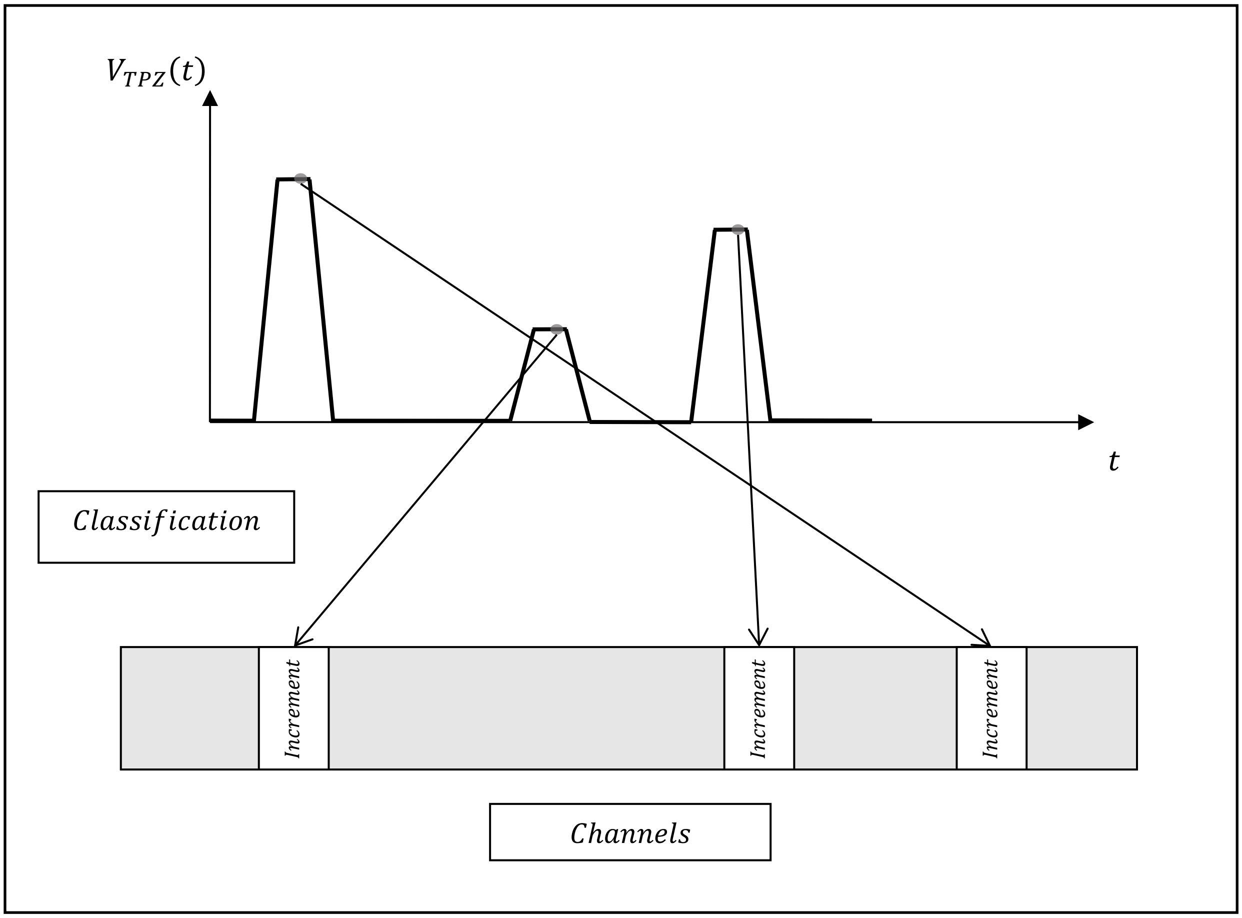


Figure II.11 The classification of the detected peaks the trapezoidal shaped pulses in the RAM.

II.10 Conclusion

In this chapter, we detailed the efficiency of the Trapezoidal filter and its advantages in ballistic deficit correction and pile up pulses to increase the energy resolution when selecting the optimal parameters of it. Some of the problems such as baseline drift and the DC offset can accrue in the baseline of the trapezoidal output; therefore, we also provided some techniques to solve those problems. In addition, we provided an algorithm for the peaks detection that will be classified by the multichannel analyzer to display the spectrum at the end.

Chapter III

Simulation of The Gamma Ray Spectroscopy Chain and Error Analysis

III.1 Introduction

In this chapter, we are going simulate the spectroscopy system going from generating the pulse coming from the detector to classifying the pulses height using MATLAB/Simulink and Xilinx System Generator Blocks that helps analyzing the system and choosing the optimal parameters for the implementation, it also makes it easy to generate a VHDL code that can be implemented in the FPGA. This chapter also includes an analysis to the digital error generated by the ADC quantization and the arithmetic operation in the FPGA to choose the optimal parameter for the FPGA implementation.

III.2 Simulation

III.2.1 Preamplifier pulse

In radiations detection, a spectroscopy system is used, which consist of a detector, preamplifier and prefilter circuit depending on the category of the digital spectrometer. The final output of this spectroscopy system is a pulse with short rise time followed by a long exponential tail which is subject to different external noises.

- To create the exponential pulse as input signal for the simulation we will need the blocks in Table III.1 and connect them as Fig III.1

Table III.1 Pulse generator subsystem blocks location.

Block location	Block name
Ports & Subsystems	Triggered Subsystem
Continuous	Zero-Pole
Sources	Pulse Generator
	Band Limited White Noise
Discrete	Memory
Commonly Used Blocks	Gain
	Scope
	Sum

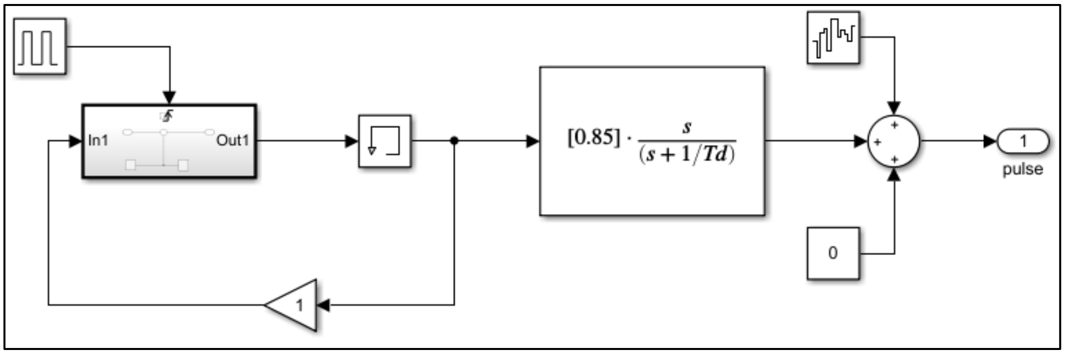


Figure III.1 Pulse generator subsystem.

- The parameters of each block are as shown in Table III.2

Table III.2 Pulse generator subsystem blocks parameters.

Block name	Parameter	Value
Pulse Generator	Amplitude	1
	Period(secs)	Tpprd*1e-6
Zero-Pole	Zeros	[0]
	Poles	[-1e+6/Td]
	Gain	[0.8]
Band-Limited white noise	Noise Power	[1e-12]
	Sample Time	delta_t
Triggered Subsystem-step	Step Time	0
	Sample Time	-1
Scope	General -Time Data	400e-6
	Data History – Limit point to	20000

1. First of all, the pulse generator will simulate the charge particle arrival time as a control signal for the Triggered subsystem (Fig III.2)

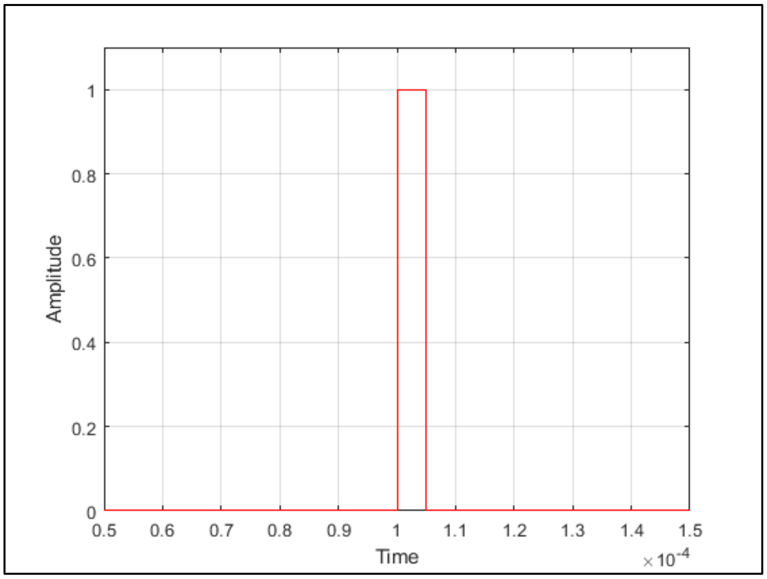


Figure III.2 The charge particle arrival time.

2. Then, “Triggered subsystem” will run a trigger signal that rises from a zero value to a positive value (Fig III.3)

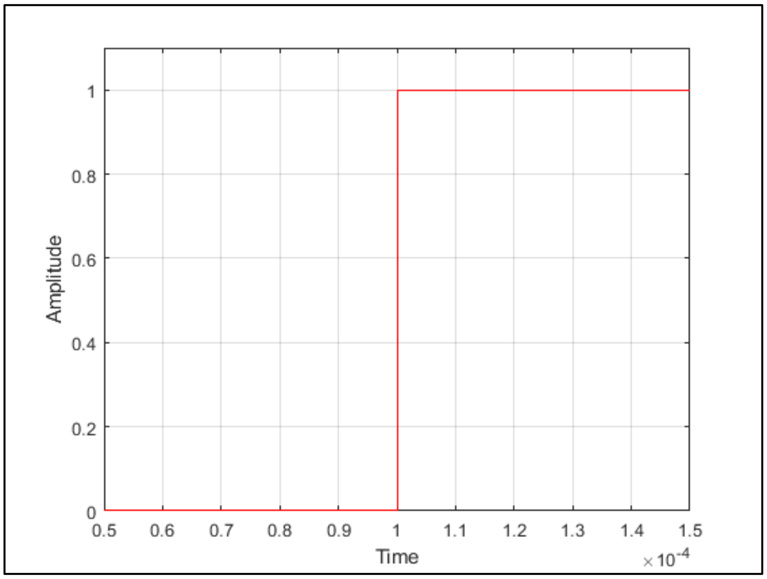


Figure III.3 Trigger signal from the triggered subsystem.

3. The Triggered subsystem, the memory and the gain will simulate the detector on which each produced unit step signal will be added to the previous value from the memory. The output from the memory will pass through a high pass filter (The Zero-pole block) with RC = Td, so the output will be a long exponential tail with Td time decay time (Fig III.4)

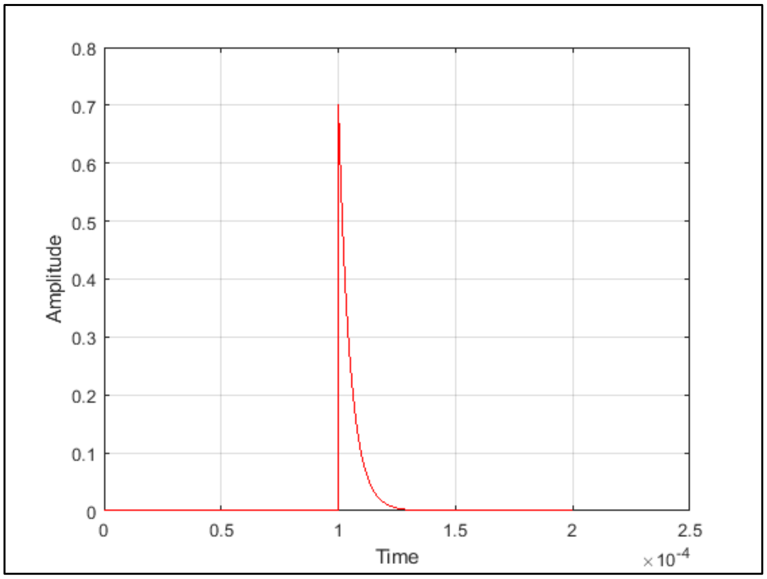


Figure III.4 Exponential decay pulse without noise.

4. We also include a white noise that will generate distributed random numbers with a specific sample rate (delta_t) (Fig III.5)

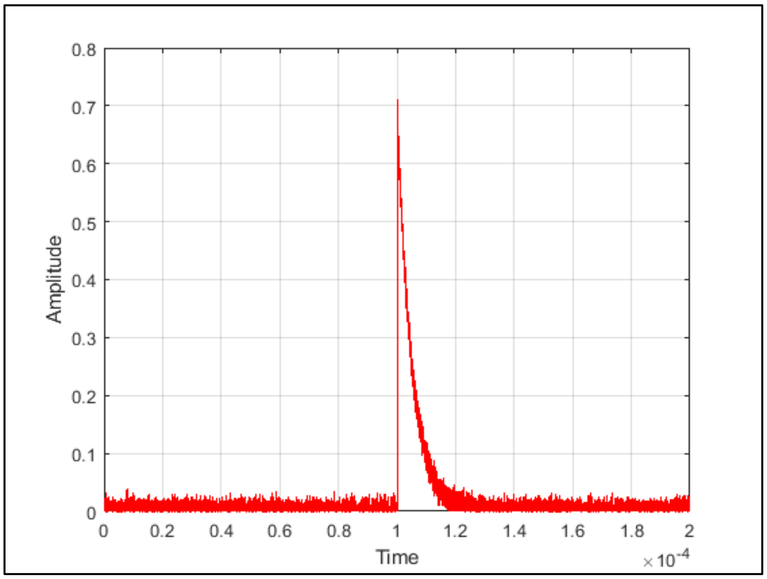


Figure III.5 Exponential decay pulse with noise.

III.2.2 ADC Quantizer

ADC is a device that provides a digital representation to the analog signal. Quantization refers to the process where an analog input value mapped to a discrete level represented by sequence of digital code (bits). The discrete level value for code level (quantization level) is dictated by the input signal, the ADC voltage reference and the resolution of ADC. Since ADC use binary code, the number of levels that it can encode is: $quantizationlevel\ = 2^{n-1}$, where n is the ADC resolution.

- The blocks in table III.3 needed to create the ADC model

Table III.3 ADC subsystem blocks location.

Block location	Block name
Simulink Extras –Additional Discrete	Idealized ADC quantizer
Simulink- Discrete	Zero order hold;
Simulink-Logic and Bit Operations:	Shift Arithmetic
Simulink-Commonly Used Blocks	Data Type conversion

- The parameters for each block will be as Table III.4

Table III.4 ADC subsystem blocks parameter.

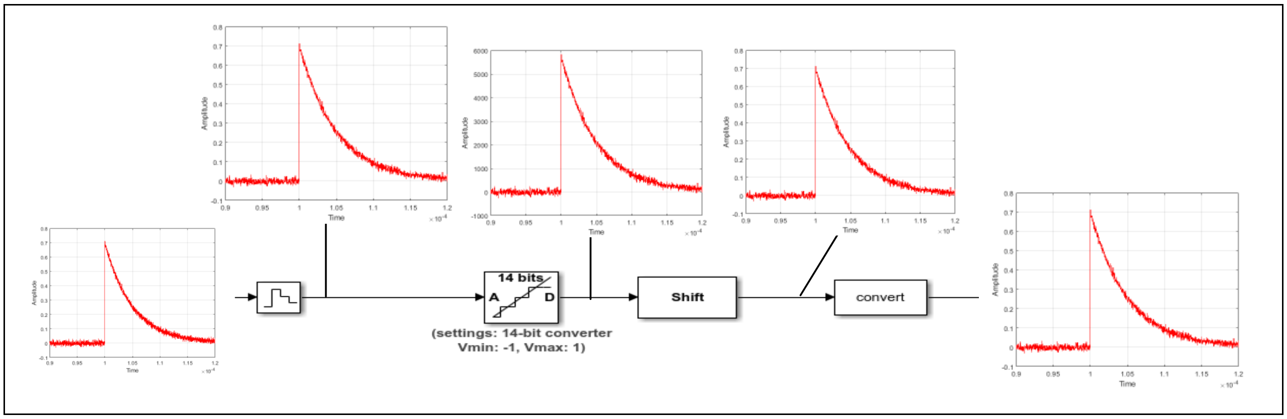
Block name	Parameter	Value
Idealized ADC quantizer	Number of converter bits	14
	Min input voltage at low output	-1
	Maxinputvoltageat2^{n}output (unreachable)	1
	Output data type	Int16
	Output negative values	Yes(checked)
Shift Arithmetic	Number of bits to shift right	0
	Number of places by which binary point shifts right	-13
Convert	Output data type	fixdt (1,14,14-1)
	Integer rounding mode	Zero
	Saturate on integer overflow	RWV
	Sample time	delta_t

- The Simulink ADC model consists of four blocks, each block performs a specific function in order to convert the real voltage input of the ADC to a digitized measured voltage. The following steps describe how the output of the ADC_Quantizer is generated

1. In the Zero-Order Hold, on each rising edge of the ADC clock the analog input signal from the pulse model is sampled and kept constant during ADC_clock period till the next rising edge (where ADC_clock sampled period of the ADC).
2. In the Idealized ADC quantizer block, the specified input voltage range of the ADC which is defined by the max and the min voltage values is split into 2^n sub-intervals. And each

subinterval is represented by a digitized values (a value in the range of $[2^{n-1}, 2^{n-1} - 1]$) where n is the resolution of the ADC.

3. Depending on which subinterval the output value of the Zero-Order Hold block belongs, it is converted to the specific digitized value that represents this interval.
4. The Shift Arithmetic block maps the digitized values of the Idealized ADC quantizer to a number that represent the measured value of the input voltage and that is done by using a simple binary shift operation.
5. The last block represents the way the numbers are represented in hardware, in this case it is represented as Signed Fixed-Point data type of 14 bits Word Length and 13 bits point and we generally use this block to improve performance of the generated code.



(settings: 14-bit converter
Vmin: -1, Vmax: 1)

Figure III.6 Different stages of the ADC Simulink model.

III.2.2.1 Error and noise due to ADC quantization:

It should be pointed that there are two quantization techniques: rounded and truncated quantization and each one of them have its own error characteristics.

The Idealized ADC use a rounding quantization method as it converts the full precision value of the input to the closest quantized number in the output. This quantization process produces information loss known as the quantization error which is defined by the error between the analog input value and its digitized value in the output and in the case of the rounded quantization this error is uniformly distributed in the interval:

$$-\frac{2^{-B_{ADC}}}{2} \le \varepsilon_{ADC} \le \frac{2^{-B_{ADC}}}{2} \qquad \text{III.1}$$

$$|\varepsilon_{ADC}| \leq \frac{2^{-B_{ADC}}}{2} \qquad \text{III.2}$$

Around a null mean value $\mu_{\varepsilon_{ADC}}$, where $B_{ADC} = 13$, is number of binary places to the right of the binary point.

For a sequence of input values, the quantization error produces a quantization noise that can be described statically and characterized by the following variance:

$$\sigma_{\varepsilon_{ADC}}{}^{2} = \frac{1}{\Delta} \int_{-\frac{2^{-B_{ADC}}}{2}}^{\frac{2^{-B_{ADC}}}{2}} (\varepsilon - \mu_{\varepsilon_{ADC}})^{2} d\varepsilon = \frac{2^{-2B_{ADC}}}{12} \qquad \text{III.3}$$

The Fig III.7 shows the produced quantization noise by the Idealized ADC :

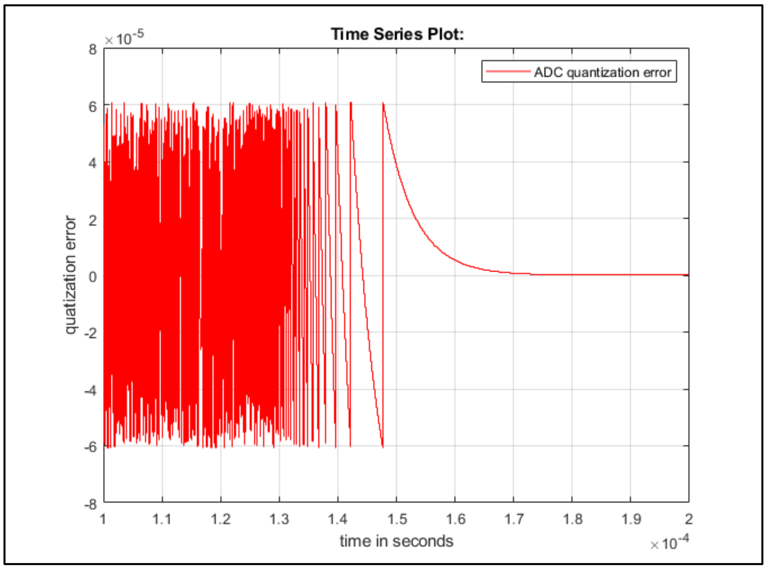


Figure III.7 The produced quantization noise by the Idealized ADC

This quantization noise by the ADC will propagate through the different stages of the filter effecting the total noise of the digital filter.

It should be noted that in order to observe the numerical errors and noises in the system, the generated white noise signal in the preamplifier pulse model is disconnected, therefore any resulted noises or error are due to numerical error by arithmetic calculation and rounding in the FPGA or by the ADC of the system.

III.2.3 Sign inverter

Real world spectroscopy pre-amplifiers have different output signal polarities (we can have preamplifiers giving only negative pulses varying from -1 to 0 or only positive pulses varying from

0 to +1). It is therefore convenient to add a digital sign inverter after the ADC that can be optionally enabled (manually) depending on the polarity of the input signal. In this case, our filter is always processing positive pulses.

The sign inverter performs a 2's complement sign inversion when the 'In' port is enable. It extracts the data type properties of the fixed-point input number and reinterpret to unsigned integer using the reinterpret function xl_force, then each bit of the resulted unsigned integer number is inverted (xl_not) and 1 is added and finally the result is converted to a Xilinx fixed-point type and reinterpret in original representation using the previous extracted properties of the input.

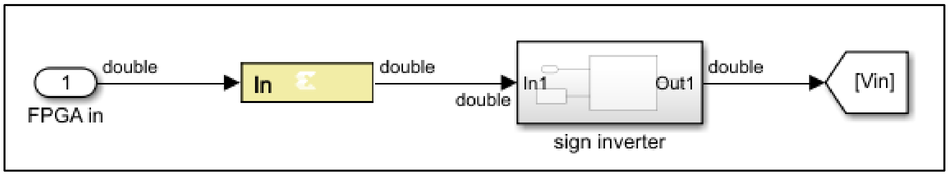


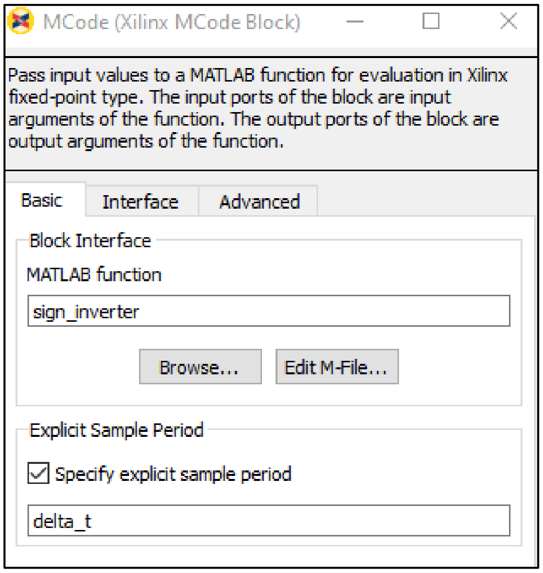
Figure III.8 Sign inverter

- The blocks in Table III.5 are needed to create sign inverter subsystem

Table III.5 Sign inverter subsystem blocks location and the Mcode script.

Block location	Block name
Xilinx Blocksets-Control Logic	MCode
Xilinx Blocksets–Basic Elements	Constant

- We browse the script to the “MCode” and change the parameters as Fig III.9 and Table III.6


MCode (Xilinx MCode Block)

Pass input values to a MATLAB function for evaluation in Xilinx fixed-point type. The input ports of the block are input arguments of the function. The output ports of the block are output arguments of the function.

Block Interface

MATLAB function: sign_inverter

Explicit Sample Period

☑ Specify explicit sample period

delta_t

Figure III.9 Mcode for the sign inverter.

Table III.6 Sign inverter subsystem blocks parameter.

Constant	Basic: Type	Unsigned
	Constant value	1
	Output Type	Boolean
	Sample constant	Yes
	Sample time	delta_t
MCode	MATLAB function	Sign_inverter
	Sample time	delta_t

- We drag the blocks and attach them as Fig III.10

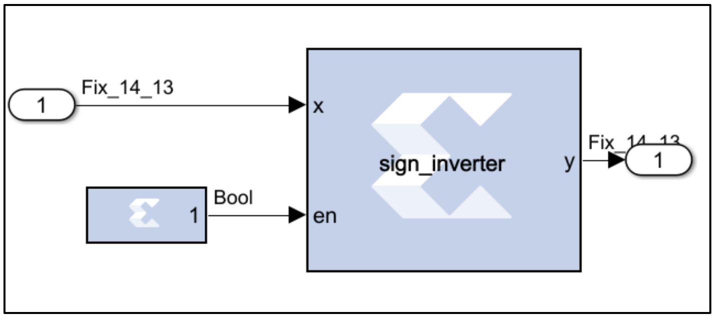


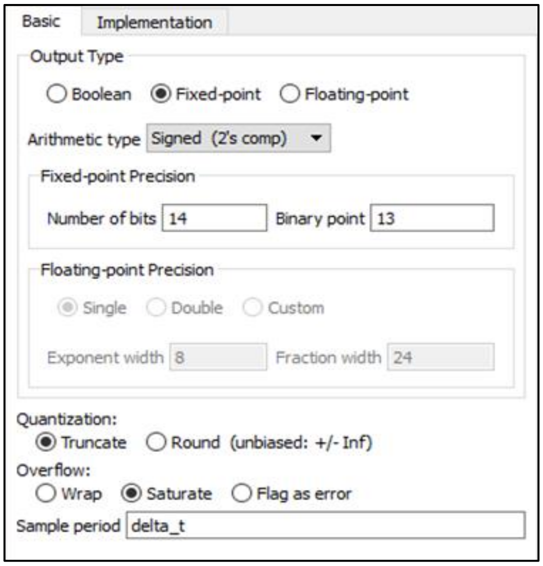
Figure III.10 Sign inverter subsystem

III.2.4 Creating filter subsystems[1]

We design the filter subsystems using Xilinx blocks instead of Simulink blocks, accordingly the input must pass by “Gateway In” and after that will pass it through four subsystems and finally the output will pass “Gateway Out”, and we need a “system generator” block to enable the development of architecture-level FPGA designs using graphical blocks programming.

- The gateway “In” parameter should change as Fig III.11

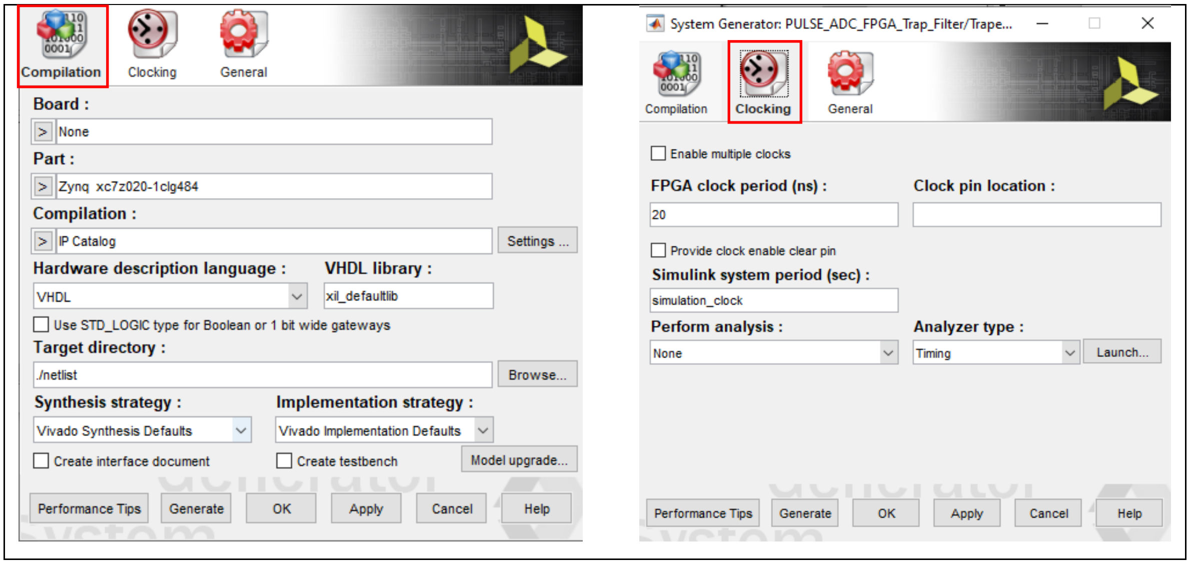
[1] The optimal choice of bits number in each Xilinx blocks for each stage that is shown in the configuration tables in this chapter is justified in the precision part for each stage.



Output Type
○ Boolean ● Fixed-point ○ Floating-point
Arithmetic type: Signed (2's comp)
Fixed-point Precision
Number of bits: 14 Binary point: 13
Floating-point Precision
● Single ○ Double ○ Custom
Exponent width: 8 Fraction width: 24
Quantization:
● Truncate ○ Round (unbiased: +/- Inf)
Overflow:
○ Wrap ● Saturate ○ Flag as error
Sample period: delta_t

Figure III.11 The gateway "In" parameter.

- The "system generator" block configuration are as Fig III.12



Board : None
Part : Zynq xc7z020-1clg484
Compilation : IP Catalog
Hardware description language : VHDL
VHDL library : xil_defaultlib
□ Use STD_LOGIC type for Boolean or 1 bit wide gateways
Target directory : ./netlist
Synthesis strategy : Vivado Synthesis Defaults
Implementation strategy : Vivado Implementation Defaults
□ Create interface document
□ Create testbench

□ Enable multiple clocks
FPGA clock period (ns) : 20
Clock pin location :
□ Provide clock enable clear pin
Simulink system period (sec) : simulation_clock
Perform analysis : None
Analyzer type : Timing

Figure III.12 "System generator" block configuration.

- Four stages which every one of them represent discrete transfer function of the filter and they are designed using the recursive formula of each transfer function based on Xilinx blocks and fixed-point arithmetic. The output of every stage linked to "Go to" block and the input of every stage is linked to "From" that make the stages in cascade form as shown in Fig III.13 box 1.

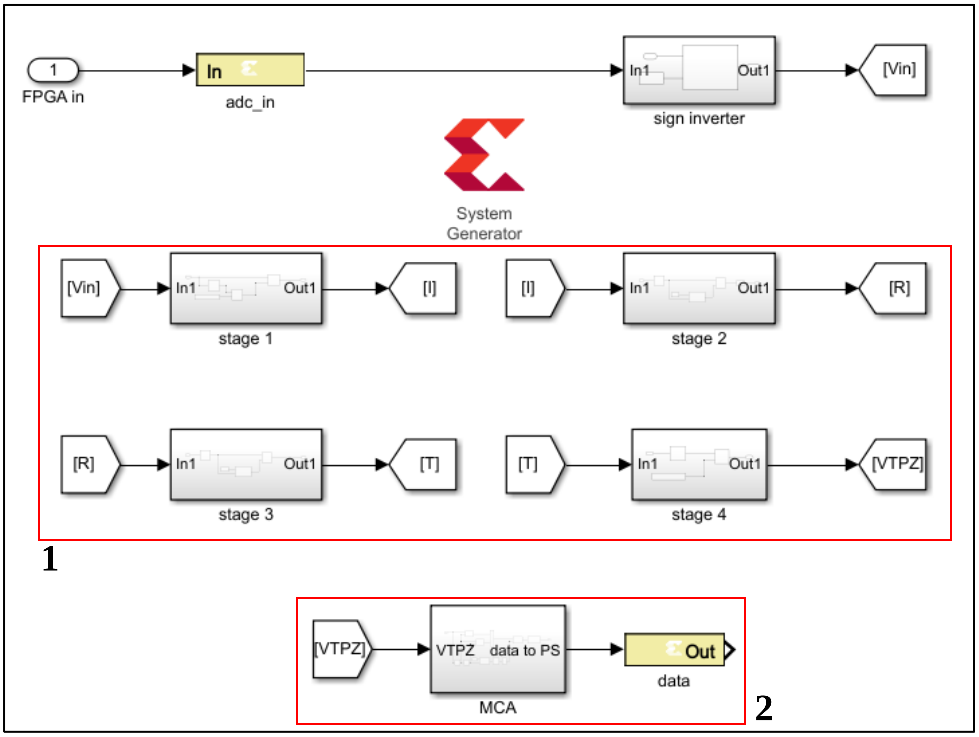


Figure III.13 Trapezoidal filter subsystem.

Table III.7 Gateway blocks location.

Block location	Block name
Simulink-Signal routing	From
Simulink-Signal routing	Go to

III.2.4.1 Stage I

The discrete transfer function of the filter of the first stage is:

$$H_{s1}(z) = H_1(z) = (1 - \beta z^{-1}) \qquad \text{III.4}$$

- The blocks needed to create stage I are shown in Table III.8

Table III.8 Stage I blocks location.

Location	Block
Xilinx Block sets–Math	Add/Subtract Mult
Xilinx Blocksets–Basic Elements	Delay Constant

- The stage subsystem is attached as Fig III.14

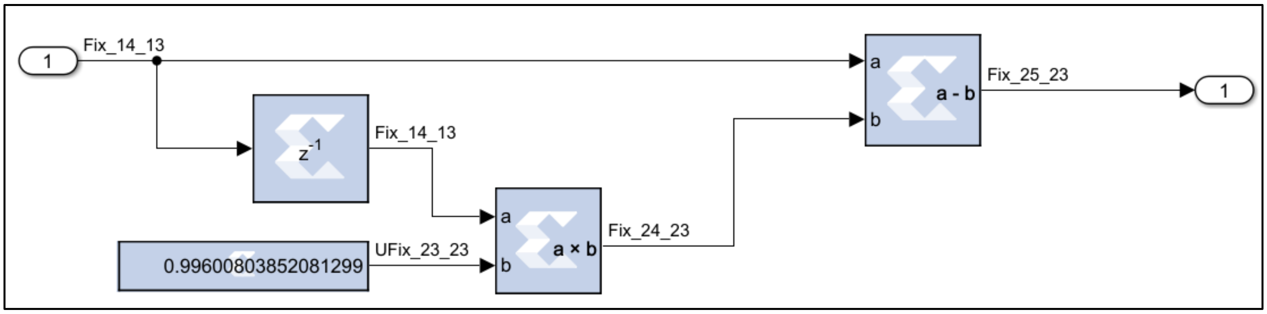


Figure III.14 Stage I subsystem

- The parameters of each block are in Table III.9

Table III.9 Stage I blocks parameter.

Block	Parameter	Value
Mult	Precision Arithmetic type Number of bits Binary point Quantization Overflow Latency	User Defined signed 24 23 Round Saturate 0
Sub	Basic : Operation Output type: Precision Latency	Subtraction Full 0
Constant	Basic: Type Constant value Number of bits Binary point Sample constant Sample time	Unsigned Beta 23 23 YES Delta_t

- The shape of the stage I output is as shown in Fig III.15

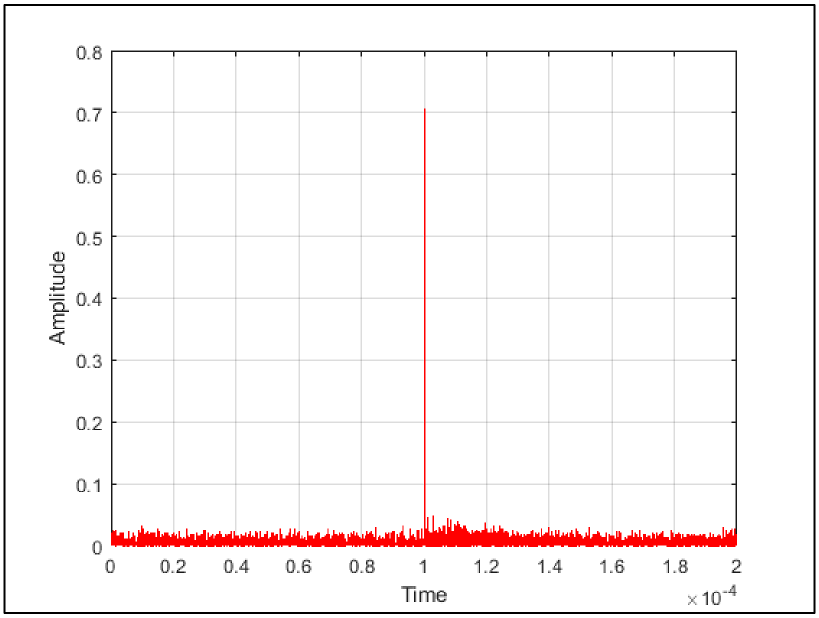


Figure III.15 Stage I output

III.2.4.2 Stage II:

The discrete transfer function of the second stage is:

$$H_{s2}(z) = \frac{H_2(z)}{H_5(z)} = \frac{(1 - z^{-A})}{(1 - z^{-1})} \qquad \text{III.5}$$

Another representation of the transfer function of the stage 2 can be obtained and it would be used later on:

$$H_{s2}(z) = \left(1 + z^{-1} + \cdots + z^{-(A-1)}\right) \qquad \text{III.6}$$

- The blocks in Table III.10 are needed to create stage 2

Table III.10 Stage II blocks location.

Location	Block
Xilinx Block set–Math	Accumulator Constant Add/Subtract
Xilinx Block set – Basic element	Convert
Delay line	RAM

- The blocks are attached as Fig III.16

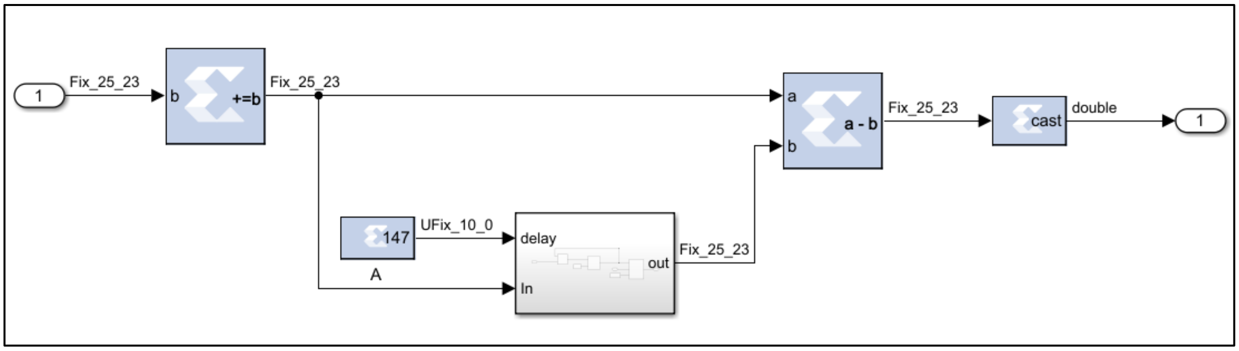


Figure III.16 Stage II subsystem.

The latency of the implemented delay line using the RAM block is 3; therefore, the value of the used delay length is A-3 instead of the delay length A to get a proper delay. Hence, we define the parameters of each block as shown in Table III.11

Table III.11 Stage II blocks parameter.

Block	Parameter	Value
Sub	Basic: Operation	Substraction
	Precision	User
	Arithmetic type	Defined signed
	Number of bits	25
	Binary point	23
	Quantization	Round
	Overflow	Wrap
Accumulator	Number of bits	25
	Overflow	Add
	Provide synchronous reset port	Unchck
	Latency	0
Constant	Basic: Type	Unsigned
	Constant value	A_delay
	Number of bits	10
	Binary point	0
	Sample constant	Check
	Sample time	Delta_t
Convert	Basic: Output Type	Fixed-point
	Precision	User
	Arithmetic type	Defined signed
	Number of bits	16
	Binary point	14
	Quantization	RoundCheck
	Overflow	Wrap

- Delay line:

The Xilinx system generator offers two blocks to construct a delay line: Delay and Addressable shift register blocks; however, both of them take a lot of resources for such long delay lines and data bytes width and they are not suitable; therefore, the efficient solution is to use internal RAM blocks in form of circular buffers.

- The location of every block we need to implement the delay line is in Table III.12

Table III.12 Delay line blocks location.

Location	Block
Xilinx Block set – Control Logic	Single Port RAM Counter
Xilinx Block set – Basic Elements	Constant Regional Block
Xilinx Block set – Index	Configure Subsystem Manager

- We will attach these blocks as Fig III.17.

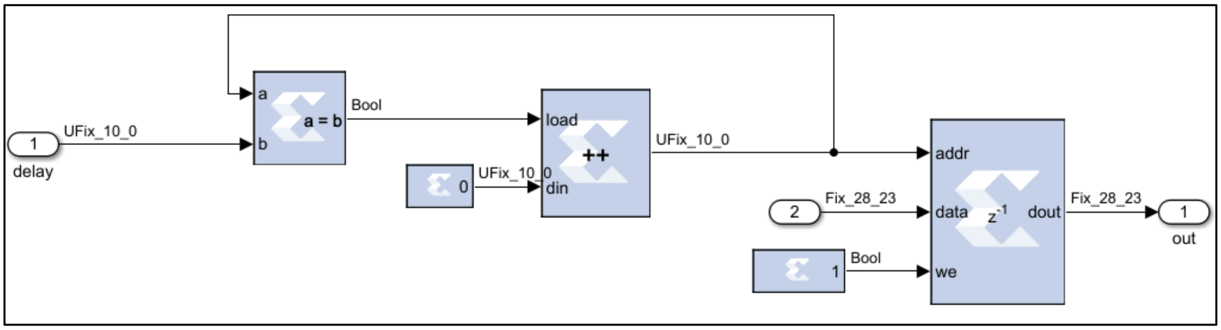
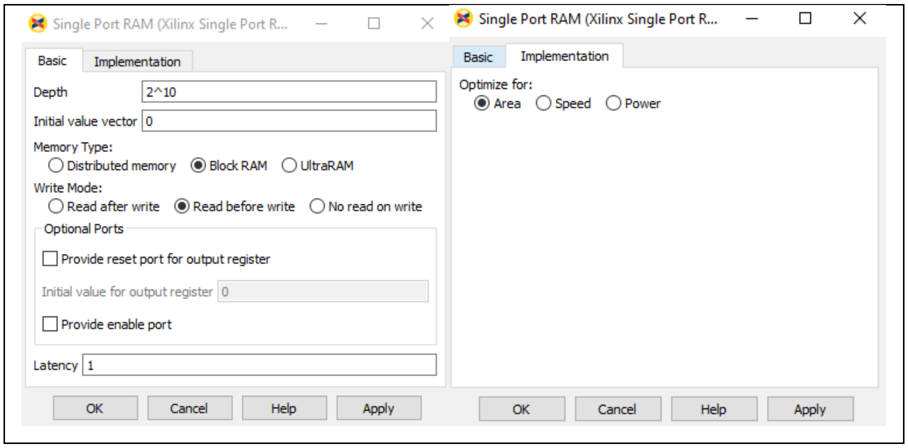


Figure III.17 Delay line subsystem.

The Single Port RAM configuration is:


Single Port RAM (Xilinx Single Port R...
Basic
Implementation
OK
Cancel
Help
Apply
Single Port RAM (Xilinx Single Port R...
Basic
Implementation
OK
Cancel
Help
Apply

Depth 2^10

Initial value vector 0

Memory Type:
○ Distributed memory ● Block RAM ○ UltraRAM

Write Mode:
○ Read after write ● Read before write ○ No read on write

Optional Ports

□ Provide reset port for output register

Initial value for output register 0

□ Provide enable port

Latency 1

Optimize for:
● Area ○ Speed ○ Power

Figure III.18 Single Port RAM configuration

Then we define the parameters of the other block as in Table III.13

Table III.13 Delay line blocks parameter.

Block	Parameter	Value
Counter	Number of bits Provide load port Explicit period	10 Yes delta_t
Constant	Basic: Type Constant value Number of bits Binary point Sample time	Boolean 1 10 0 delta_t
Convert	Basic: Output Type Precision Arithmetic type Number of bits Binary point Quantization Overflow	Fixed-point User Defined signed 16 14 Round Wrap
Constant	Basic: Type Constant value Number of bits Binary point Sample time	Fixed point 0 10 0 delta_t
Regional Block	All default	
Configure Sybsystem Manager	All default	

- The shape of the stage II output is as shown in Fig III.19

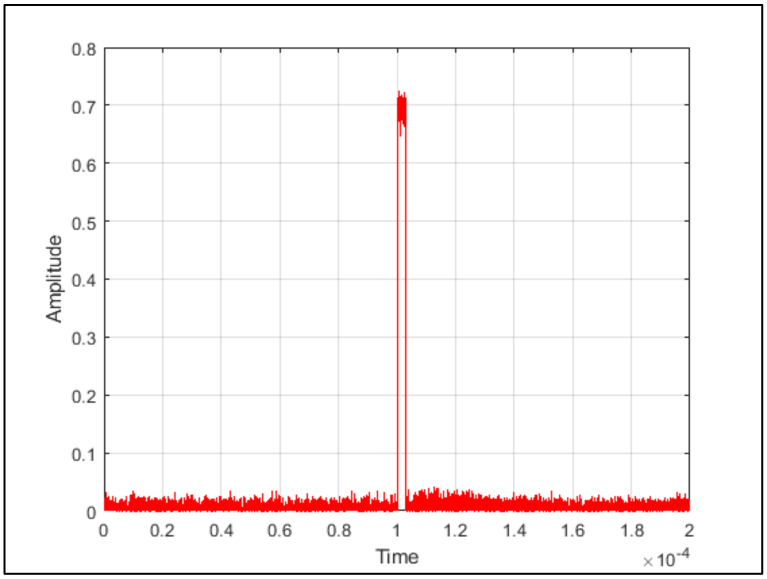


Figure III.19 Stage II output

III.2.4.3 Precision of stages I and II[2]:

It is important to consider the finite word-length limitation of the hardware when implementing a digital filter therefore as the filter coefficient β can not be represented in full precision, it is quantized to a finite number of bits that will change the frequency response characteristics of the filter and produce additional error ε_{β,I_n} in the output of stage I. We denote ε_β the rounding quantization error of the coefficient β and therefore the recursive formula of the input of stage I can be written as:

$$I_n = V_{IN_n} - (\beta + \varepsilon_\beta) V_{IN_{n-1}} \qquad \text{IIII.7}$$

β here represents the rounded value of the real β and that result $(\beta + \varepsilon_\beta)$ in the above equation represent the real value of β so that:

$$\Rightarrow I_n = V_{IN_n} - \beta V_{IN_{n-1}} - \varepsilon_\beta V_{IN_{n-1}} \qquad \text{IIII.8}$$

$$\Rightarrow \varepsilon_{\beta,I_n} = -\varepsilon_\beta V_{IN_{n-1}} \qquad \text{IIII.9}$$

$$for\ n \geq 0$$

And by neglecting the rise time of the input signal and taking$E = 1$, $V_{IN_{n-1}}$for $n \geq 0$can be written as:

$$V_{IN_{n-1}} = e^{-\frac{\Delta t}{\tau_d}(n-1)}, n \geq 0 \qquad \text{III.10}$$

By replacing in we found:

$$\varepsilon_{\beta,I_n} = -\varepsilon_\beta e^{-\frac{\Delta t}{\tau_d}(n-1)} \qquad \text{III.11}$$

This error ε_{β,I_n} in the output of stage I will produce an error ε_{β,R_n} in the output of the stage $II$and that make the output of stage II can be written as:

$$R_n = I_n + I_{n-1} + \cdots + I_{n-(A-1)} \qquad \text{IIII.12}$$

The error ε_{β,R_n}can up to a maximum error of:

$$\left| max\ (\varepsilon_{\beta,R_n}) \right| = \left| \sum_{i=0}^{A-1} \varepsilon_{\beta,I_n} \right| \approx A \varepsilon_\beta e^{\frac{\Delta t}{\tau_d}} \qquad \text{IIII.13}$$

[2] The analysis of quantization error and its calculations was based on [24]

As $A = \frac{t_1}{\Delta t}$ and by taking peaking time $t_1 = 3us$ and $\Delta t = 10ns$, it leads to that $A < 2^{10}$.Also because of that $\tau_d = 5us$ we get:

$$\left| max\ (\varepsilon_{\beta,R_n}) \right| \approx 2^{10}\varepsilon_\beta \qquad \text{IIII.14}$$

By taking that the required precision to be $\left| max\ (\varepsilon_{\beta,R_n}) \right| < 2^{-13}$ we get

$$2^{-13} > 2^{10}\varepsilon_\beta \qquad \text{IIII.15}$$

Therefore, the condition for $\varepsilon_\beta$is:

$$\varepsilon_\beta < 2^{-10}2^{-13} \qquad \text{III.16}$$

$$\varepsilon_\beta < 2^{-23} \qquad \text{III.17}$$

This imposes that β must be represented with at least 23 bits as the maximum rounding error of it will be:

$$max\ (\varepsilon_\beta) = \frac{2^{-23}}{2} < 2^{-23} \qquad \text{III.18}$$

Hence, the format of β must bea 23.23 since that $\beta < 1$.

III.2.4.3.1Propagation of the ADC quantization error through stages I and II:

The quantization noise produced by the ADC quantization error will popagate through stages and effects the output of each stage. The total trasfer function of stage I and II will be used to determine the variance of that propagated noise in the output of stage II.

$$H_{s1}(z)H_{s2}(z) = (1 - \beta z^{-1})\left(1 + z^{-1} + \cdots + z^{-(A-1)}\right) \qquad \text{III.19}$$

$$H_{s1}(z)H_{s2}(z) = 1 + z^{-1} + \cdots + z^{-(A-1)} - \beta z^{-1} - \beta z^{-2} - \cdots - \beta z^{-A} \qquad \text{III.20}$$

$$H_{s1}(z)H_{s2}(z) = 1 + (1-\beta)z^{-1} + (1-\beta)z^{-2} + \cdots + (1-\beta)z^{-(A-1)} - \beta z^{-A} \qquad \text{III.21}$$

And can also be written as:

$$H_{s1}(z)H_{s2}(z) = \sum_{k=0}^{A} h_k z^{-k} \qquad \text{III.22}$$

Assuming that the errors are independent, the propagated variance at the output of stage II due to ADC quantization error is:

$$\sigma_{\varepsilon_{ADC},s2}{}^2 = \sigma_{\varepsilon_{ADC}}{}^2 \sum_{k=0}^{A} h_k{}^2 \approx \sigma_{\varepsilon_{ADC}}{}^2 \left(A \left(\frac{\Delta t}{\tau_d} \right)^2 + 2 \right) \approx 2\sigma_{\varepsilon_{ADC}}{}^2 = 2\frac{2^{-2B_{ADC}}}{12} \qquad \text{IIII.23}$$

III.2.4.3.2Error due to arithmetic operations in stage I and II:

- An arithmetic operation noise is produced in stage I and II, thus after each multiplication in stage I the result is rounded producing an error that is accumulated by the accumulator in stage II resulting to an arithmetic noise that is characterized by the noise variance $\sigma_{accumulator}{}^2$ which is A times larger than variance due to single multiplication in stage $I\sigma_{mult,s1}{}^2$.

$$\sigma_{accumulator}{}^2 = A.\sigma_{mult,s1}{}^2 = A\frac{2^{-2B_{mult}}}{12} \qquad \text{III.24}$$

Where $B_{mult}$is the precision of the multiplication block in stage I.

- Therefore, the variance of the total noise at the output of the second stage is the sum of the ADC quantization noise variance and the variance of the sum-of-product noise as shown below:

$$\sigma_{s2}{}^2 = \sigma_{\varepsilon_{ADC},s2}{}^2 + \sigma_{accumulator}{}^2 = 2\frac{2^{-2B_{ADC}}}{12} + A\frac{2^{-2B_{mult}}}{12} \qquad \text{III.25}$$

- And by selecting an appropriate optimal value of $B_{mult} = 23$ we can neglect the noise produced by the arithmetics operation in stage I and II and $\sigma_{s2}{}^2$ can be approximated by:

$$\sigma_{s2}{}^2 = 2\frac{2^{-2B_{ADC}}}{12} \qquad \text{III.26}$$

III.2.4.3.3Gain of stage I and II at DC frequency:

At DC frequency the input V_{IN} is a constant $V_{IN} = cnst$, then:

$$I_n = V_{IN_n} - \beta V_{IN_{n-1}} \qquad \text{III.27}$$

$$I_n = cnst - \beta cnst \qquad \text{III.28}$$

$$I_n = cnst(1-\beta) \qquad \text{III.29}$$

Therefore, the gain at DC frequency in stage I is:

$$I_n = cnst.G_{s1} \qquad \text{III.30}$$

$$G_{s1} = 1 - \beta = 1 - e^{-\frac{\Delta t}{\tau_d}} \approx \frac{\Delta t}{\tau_d} \tag{III.31}$$

And as $\Delta t = 10ns$and $\tau_d = 5us$, therefore $G_{s1} = 0.002$. This has advantage that any DC offset at the output of the stage Iwill be very low.

Recall that the corresponding recursive relation of stage II is:

$$R_n = I_n + I_{n-1} + \cdots + I_{n-(A-1)} \tag{III.32}$$

In the case of DC frequency,$I \quad = cnst$:

$$R_n = cnst + cnst + \cdots + cnst \tag{III.33}$$

$$R_n = A \times cnst \tag{III.34}$$

Which means that the gain G_{s2} at DC frequency in stage $II$is equal to:

$$G_{s2} = A = \frac{t_1}{\Delta t} \tag{III.35}$$

The total gain after the second stage at DC is:

$$G_{s1s2} = G_{s1}.G_{s2} = \frac{\Delta t}{\tau_d}.\frac{t_1}{\Delta t} = \frac{t_1}{\tau_d} = \frac{3us}{5us} = 0.6 \tag{III.36}$$

Since the input signal is in the interval ±1, and the gain is 0.6, the output from the stage II can be represented by 2 bits to the left side of the binary point (1 bits for value and 1 bit for sign). Since the precision must be 23 bits to the right of the decimal point (as required by noise propagation), the stage II requires 25-bitaccumulator and 25.23-bit subtractor in order to avoid overflows and decrease effect off error propagation to tolerable level.

After all operations in the stage I and II are done in precision required by noise propagation and overflow suppression, it is possible to round result down to 14 bits to the right of the binary point, that is to output format of 16.14. This will save FPGA resources and it can be justified on the following way. The variance of error introduced by the above rounding is given by the relation:

$${\sigma_{convet,I}}^2 = \frac{2^{-2\times14}}{12} \tag{III.37}$$

The total variance at output of the stage is:

$$\sigma_{s2}{}^{2} = 2\frac{2^{-B_{ADC}}}{12} + \frac{2^{-2\times 14}}{12} = \frac{2^{-B_{ADC}}}{12}(2+\frac{1}{4}) \approx 2\frac{2^{-B_{ADC}}}{12} \qquad \text{III.38}$$

In order to decrease output precision, add Converter block from Xilinx Block set / Mat. Specify the parameter as Table III.14

Table III.14 Convert block parameter.

Convert	Number of bits	16
	Binary point	14
	Quantization	Round
	Overflow	Wrap

III.2.4.4 Stage III:

The discrete transfer function of the filter of the first stage is:

$$H_{s3}(z) = \frac{H_3(z)}{H_6(z)} = \frac{\left(1 - z^{-(A+B)}\right)}{(1 - z^{-1})} \qquad \text{III.39}$$

Another representation of the transfer function of the stage III can be obtained and it would be used later on:

$$H_{s3}(z) = \left(1 + z^{-1} + \cdots + z^{-(A+B-1)}\right) \qquad \text{III.40}$$

- We implement the third stage same as the second but without the convert block as in Fig III.20

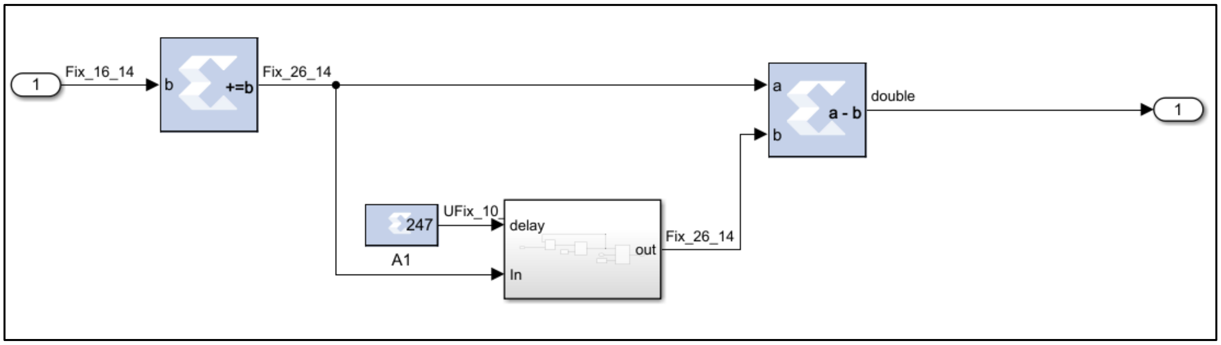


Figure III.20 Stage III subsystem

- We define the parameters of each block as follow:

Table III.15 Stage III blocks parameter.

Block	Parameter	Value
Add/Sub	Basic: Operation Precision Arithmetic type Number of bits Binary point Quantization Overflow	Substraction User Defined signed 26 14 Round Wrap
Accumulator	Number of bits	26
Constant	Basic: Type Constant value Number of bits Binary point Sample constant Sample time	Unsigned AplusB_delay 10 0 Check delta_t
Delay	Same as stage II	

Because of that, the latency of the implemented delay line using the RAM block is 3, the value of the used delay length is A+B-3 instead of the delay length A+B in order to get proper delay.

- The shape of the output is as shown in Fig III.21

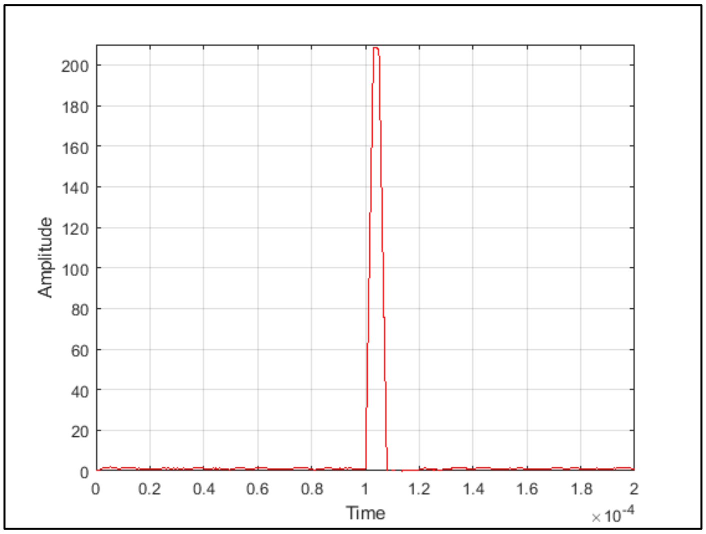


Figure III.21 Stage III output.

III.2.4.5 Stage IV

The discrete transfer function of the filter of the first stage is:

$$H_{s4}(z) = H_4(z) = \frac{z^{-1}}{A} \tag{III.41}$$

The blocks in Table III.17 are needed to implement the second stage

Table III.16 Stage IV block location.

Location	Block
Xilinx Block sets–Math	Mult
Xilinx Block sets–Basic Elements	Delay Constant

- We will attach the block like this:

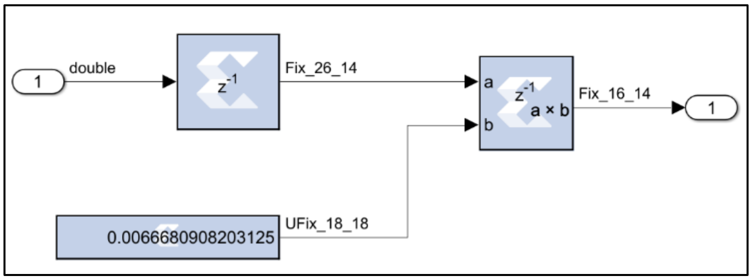


Figure III.22 Stage IV subsystem.

- The parameters of each block are defined as in Table III.18

Table III.17 Stage IV block parameter.

Block	Parameter	Value
	Precision	User
	Arithmetic type	Defined signed
	Number of bits	16
	Binary point	14
	Quantization	Round
	Latency	1
	Overflow	Saturate
Constant	Basic: Type	Unsigned
	Constant value	1/A
	Number of bits	18
	Binary point	18
	Sample constant	Check
	Sample time	delta_t
Delay	Latency	1

- The shape of the stage IV output is as shown in Fig III.23

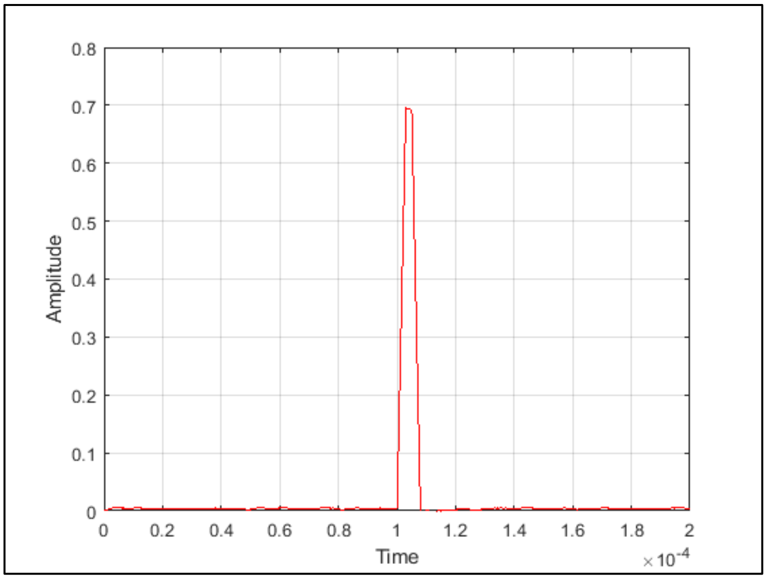


Figure III.23 Stage IV output

The Fig III.24 shows the input pulse versus the final output of the digital trapezoidal shaper

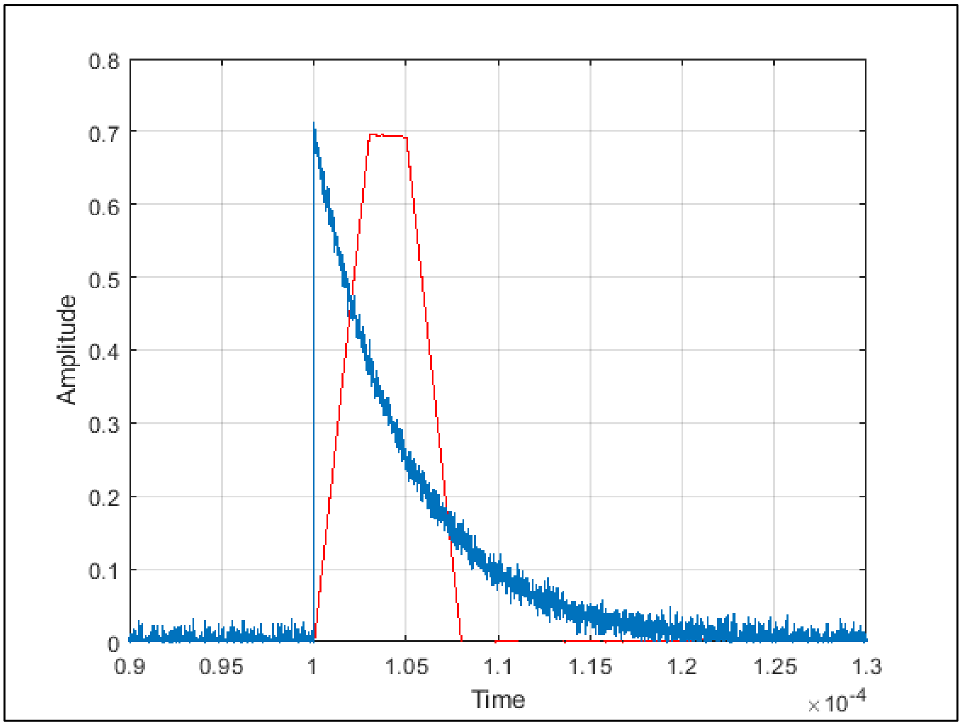


Figure III.24 The input pulse versus the final output of the DTS.

III.2.4.6 Precision of stages III and IV:

As the stage IV contains only summation which is done in full precision, the only produced noise in the output is the propagated noise from the output of the second stage. And as the total approximated total noise variance at the output of the stage $II$is:

$$\sigma_{s2}{}^{2} = 2\frac{2^{-2B_{ADC}}}{12} \tag{III.42}$$

And the discrete transfer function of the third stage is given by:

$$H_{s3}(z) = \left(1 + z^{-1} + \cdots + z^{-(A+B-1)}\right) = \sum_{k=0}^{A+B-1} h_k z^{-k} \tag{III.43}$$

The variance in the output of the third stage can be written as:

$$\mathrm{\sigma}_{\mathrm{s3}}{}^{2} = \mathrm{\sigma}_{\mathrm{s2}}{}^{2} \sum_{\mathrm{k=0}}^{\mathrm{A+B-1}} \mathrm{h_k}{}^{2} = \mathrm{\sigma}_{\mathrm{s2}}{}^{2} \sum_{\mathrm{k=0}}^{\mathrm{A+B-1}} 1 = (\mathrm{A} + \mathrm{B})\mathrm{\sigma}_{\mathrm{s2}}{}^{2} = 2(\mathrm{A} + \mathrm{B})\frac{2^{-2\mathrm{B_{ADC}}}}{12} \tag{III.44}$$

The stage IV contains only one factor that is $\frac{1}{\mathrm{A}}$ in its discrete transfer function therefore the propagated overall noise to the output of the stage IV which is the output of the filter will have the variance of:

$$\mathrm{\sigma}_{\mathrm{s4}}{}^{2} = \frac{2(\mathrm{A} + \mathrm{B})}{\mathrm{A}^2}\frac{2^{-2\mathrm{B_{ADC}}}}{12} \tag{III.45}$$

III.2.4.6.1Gain of stage II and IV at DC frequency:

Recall that the corresponding recursive relation of stage III is:

$$\mathrm{T_n} = \mathrm{R_n} + \mathrm{R_{n-1}} + \cdots + \ \mathrm{R_{n-(A+B-1)}} \tag{III.46}$$

By assuming that the input is constant R $= \mathrm{cnst}$:

$$\mathrm{T_n} = \mathrm{cnst} + \mathrm{cnst} + \cdots + \mathrm{cnst} \tag{III.47}$$

$$\mathrm{T_n} = (\mathrm{A} + \mathrm{B}) \times \mathrm{cnst} \tag{IIII.48}$$

Which means that the gain $\mathrm{G_{s3}}$ at DC frequency in stage III is equal to:

$$\mathrm{G_{s3}} = \mathrm{A} + \mathrm{B} = \frac{\mathrm{t_2}}{\Delta\mathrm{t}} = \frac{5\mathrm{us}}{10\mathrm{ns}} = 500 \tag{IIII.49}$$

Because of that the output of the stage II is 16.14 bits format and that $(\mathrm{A}+\mathrm{B}) < 2^{10}$, the stage III should have an accumulator length that is 10 bits wider then output from the stage II accordingly it would have width 26 bits and therefore the subtractor format would be 26.14 in order to keep the precision of the subtractor 14 bit to the right of the binary point.

Table III.18 Effect of A+B value in the parameters of stage III blocks.

	With DC removal in the stage 2 output	
Maximal value of $\mathrm{A}+\mathrm{B}$	Stage 3 Acc width[bits]	Stage 3 Sub format[bits]
$(\mathrm{A}+\mathrm{B}) < 2^{10}$	16+10	(16+10).14
$(\mathrm{A}+\mathrm{B}) < 2^{9}$	16+9	(16+9).14
$(\mathrm{A}+\mathrm{B}) < 2^{8}$	16+8	(16+8).14

For stage II, the gain $\mathrm{G_{s4}}$ at DC frequency is:

$$\mathrm{G_{S4}} = \frac{1}{\mathrm{A}} = \frac{\Delta \mathrm{t}}{\mathrm{t_1}} \qquad \text{III.50}$$

Therefore, the total gain $\mathrm{G_{TPZ}}$ at DC frequency of the DTS filter is:

$$\mathrm{G_{TPZ}} = \mathrm{G_1}.\mathrm{G_2}.\mathrm{G_3}.\mathrm{G_4} = \left(\frac{\Delta \mathrm{t}}{\tau_\mathrm{d}}\right).\left(\frac{\mathrm{t_1}}{\Delta \mathrm{t}}\right).\left(\frac{\mathrm{t_2}}{\Delta \mathrm{t}}\right).(\frac{\Delta \mathrm{t}}{\mathrm{t_1}}) = \frac{\mathrm{t_2}}{\tau_\mathrm{d}} \qquad \text{III.51}$$

In our case $\mathrm{t_2} = \tau_\mathrm{d} = 5\mathrm{us}$ it follows:$\mathrm{G_{TPZ}} = 1$.

Therefore, any DC offset at the input of the stage I is amplified by a factor of 1.

III.2.5 Maxima calculation and multichannel analyzer

The MCA subsystem (Fig III.13 box 2) performs the maxima detection, counting, classifying the maxima and sending a packet of the result to a processing system. It consists of 4 parts that perform different tasks (Fig III.25)

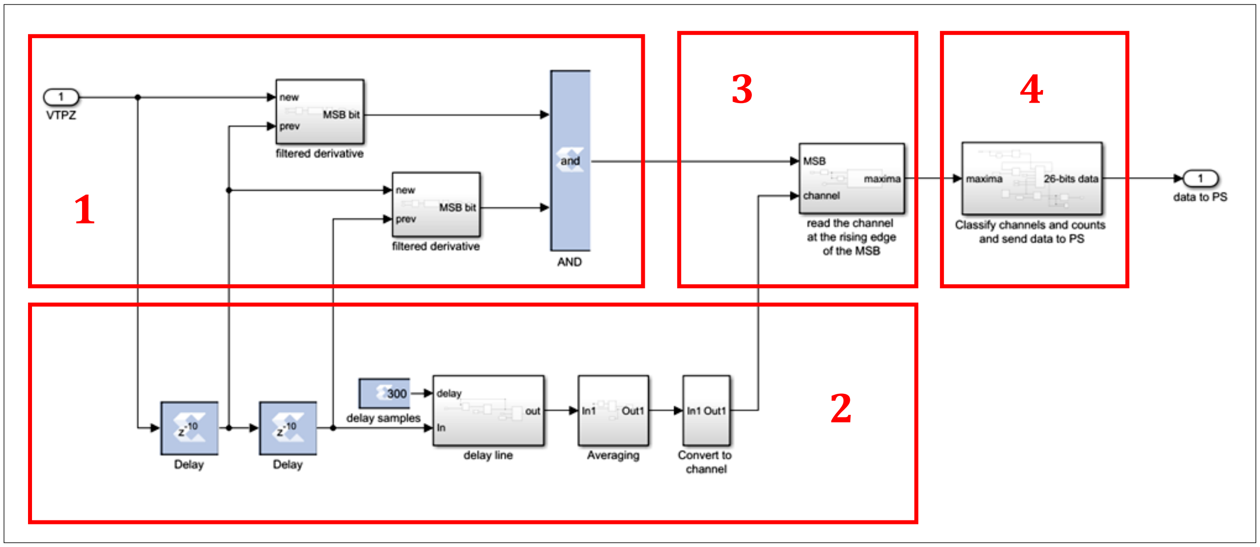


Figure III.25 The MCU subsystem

III.2.5.1 Part 1

The same technique described in chapter II is used to detect the maxima value with some simple changes, for simplicity reasons we select: $n_1 = 2$, $n_2 = 300$ and $n_3 = 2$.

The first part calculates the first derivative of the trapezoidal filter and extracts the MSB bit which will be 1 when the derivative is negative; therefore, it detects the fall of the trapezoidal output.

As shown in part 1 in Fig III.25, this operation is performed twice and combined with the used logical AND operation just to decrease the chance of having negative fault detection caused by noise.

The derivative subsystem in part 1 extracts the MSB bit of the low pass filtered difference between two samples delayed each other by 10 samples as shown in Fig III.25

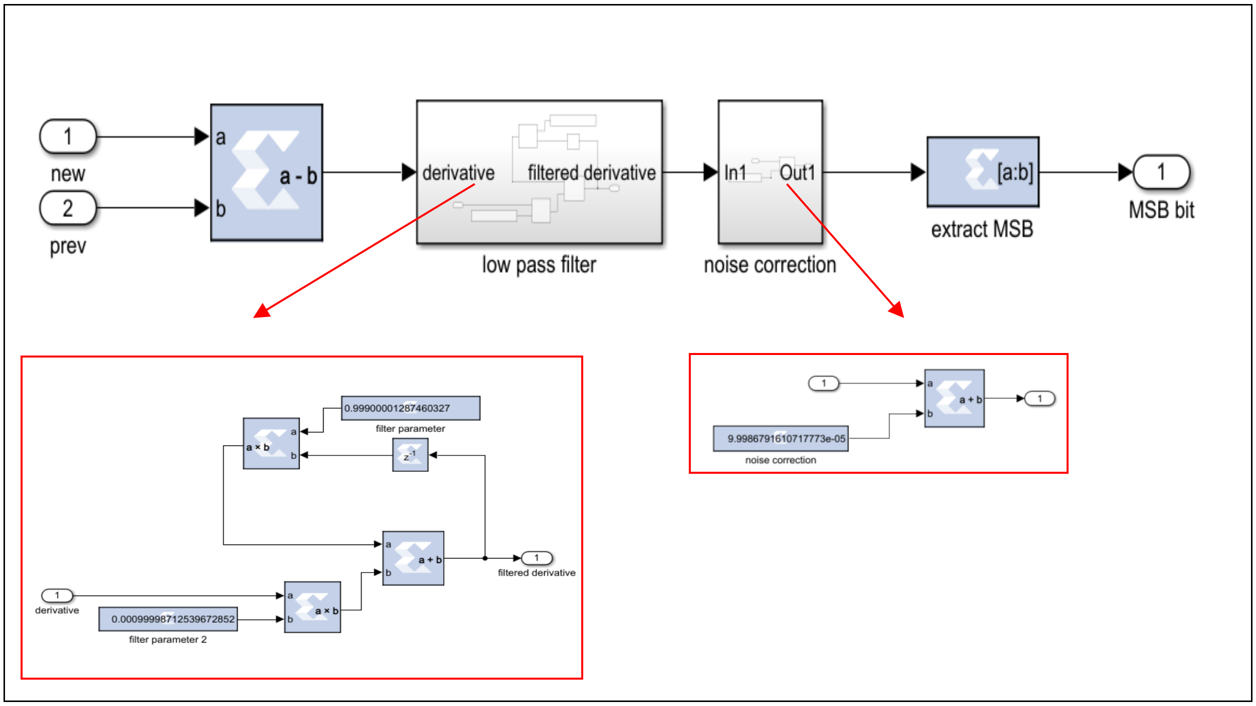


Figure III.26 The derivative subsystem

The derivative is in fact the difference between two successive samples; however, here it was performed as the difference between two samples spread by 10 samples instead of one for the same reason of using the low pass filter.

The filtered derivative subsystem includes two parameters that are: a that is the low filter parameter, and c that represent the offset that compensate the negative derivative values caused by the noise during the flat top, they were chosen by adjusting them while fixing a small amplitude pulse in the input ($a = 0.001$ and $c = 0.0001$ for the simulation).

III.2.5.2 Part 2

It calculates the average of each two successive samples in the 320-samples delayed trapezoidal output and converts the result to a channel form by taking only the integer value of the multiplication of the averaging result by 1023, giving a trapezoidal output at its output that have only integer values between 0 and 1023.

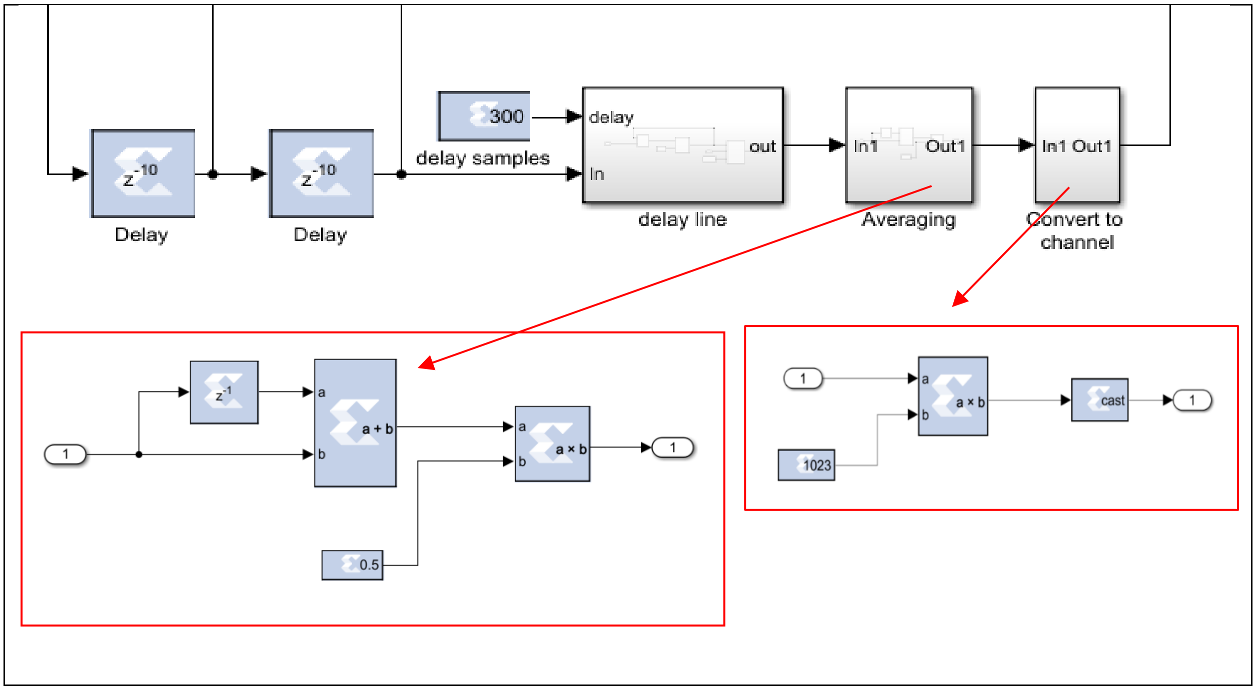


Figure III.27 Averaging and converting blocks.

III.2.5.3 Part 3

It outputs the value of the output of part 2 for a one clock cycle duration and only when it detects a rising edge from the output of part 1 and zero otherwise which result a small rect pulse that its height is the calculated maxima of the input signal to the MCU subsystem.

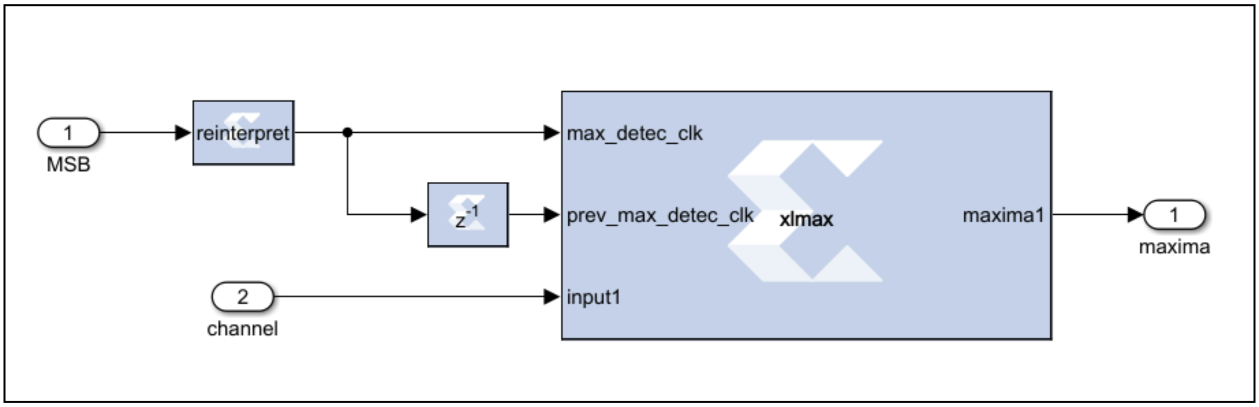


Figure III.28 Read the channel at the rising edge of the MSB blocks.

III.2.5.4 Part 4

Basically, it uses a dual port RAM where port A is used for counting however port B is only used to read the values and send them to the processing system part.

It uses the created rect pulse by part 3 that is the maxima value as an address for a dual port RAM and it increment the stored value at this address by one at each time the address is selected in that way the addresses represent the channels and the stored values represent the counts of each channel

The delay with the summation block in the output of part 3 are used to stabilize the address of port A during the read and write from port A that requires two clock cycle.

The port B address is connected to a counter that select the address to read at each time which is the channel and the output of port B is the count at the selected address.

The Bit Basher Xilinx block is used to mix the reader channel and its corresponding counts value in order to be sent to the processing part in form of one package consist of 26 bits (10bits for the channels and 16 bits for the counts) that will be processed later on to extract the channel and the counts by knowing the bits length of each one.

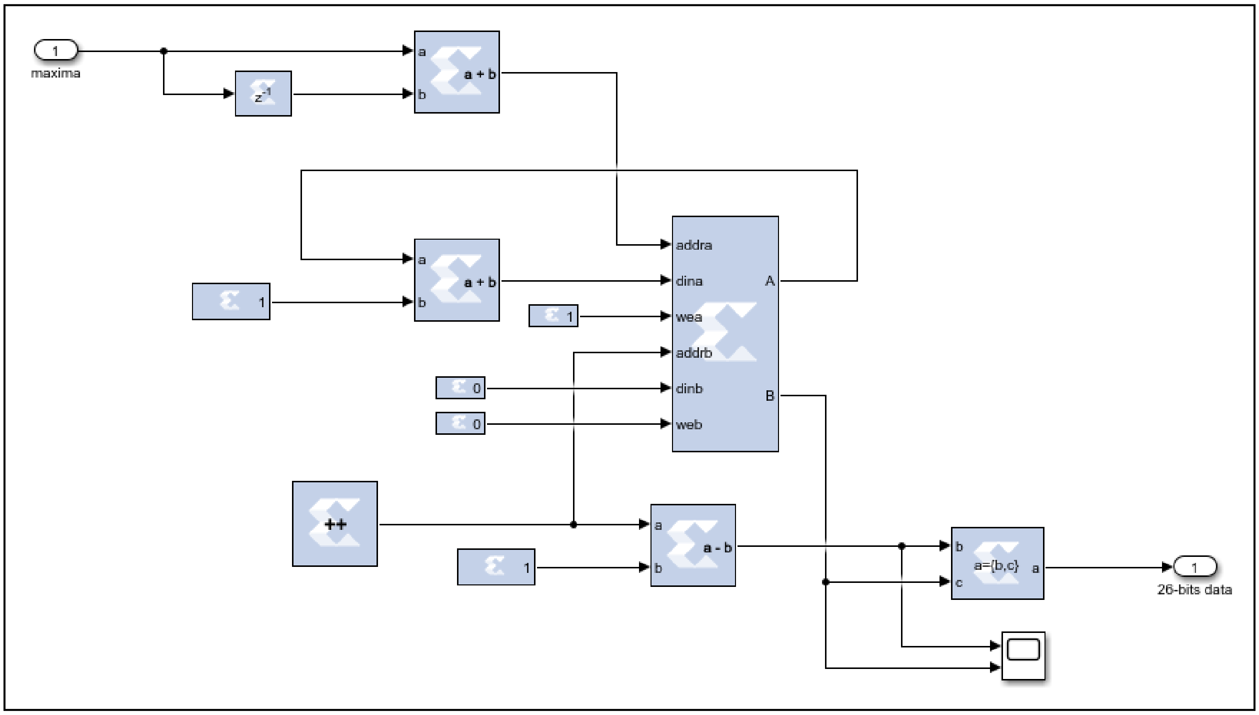


Figure III.29 Classifying channels and sending data to the PS blocks.

III.3 Conclusion

After simulating the spectroscopy system by generating the pulse coming from the detector and the preamplifier to classifying the pulses height using MATLAB/Simulink and Xilinx System Generator Blocks, a full analysis of the system to choose the optimal parameters for the implementation was needed in such a way that the error at the output is mostly caused by the quantization error of the ADC, it also makes it easy to generate a VHDL code that can be implemented in the FPGA after we are satisfying with the design and add it in form of IP in the Vivado software to be implemented later in the FPGA.

Chapter IV

Implementation of the DTF with Multichannel Analyzer and The Spectrum Display System

IV.1 Introduction

In this chapter, we will implement what we have simulated in the previous chapter and also include more operations to solve some of the problems that occur during the real implementation. We generate a series of exponential decay pulses that simulate the output of the preamplifier preceded by the HPGe detector and then digitize it with a fast ADC to fetch it into the Zedboard that processes it, to send the data obtained to a laptop in order to display the spectrum in real time and as software, the generated IP by MATLAB with Vivado Design Suite with IP Integrator and software Development Kit (SDK) is needed to program the zynq7000 System-on-Chip (SoC). By combining both PS and PL units of the chip we can achieve the most efficient design for implementing our project. Then we also designed an application in laptop using Processing software that would read the received data from the com port and display the spectrum on the screen.

IV.2 Experimental 1st setup

We generate the exponential decay pulse using Dspace and control its amplitude using a potentiometer then pass it through a fast ADC, that is attached to our Zedboard which will perform the signal processing and detect the peaks to classified them and display the spectrum at the end in the PC. The total steps for the test are shown in Fig IV.1 and the setups is in Fig IV.2

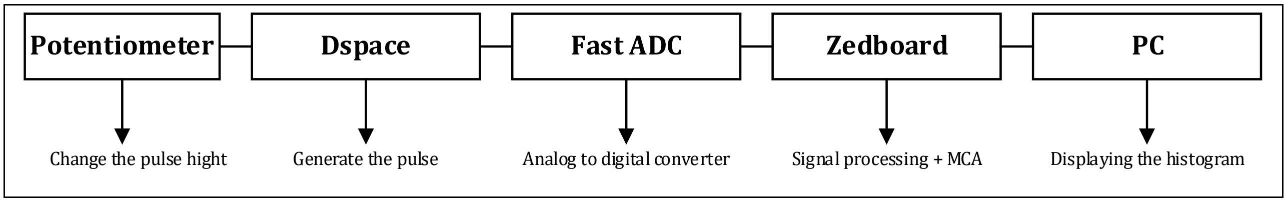


Figure IV.1 The general block diagram of the used setup.

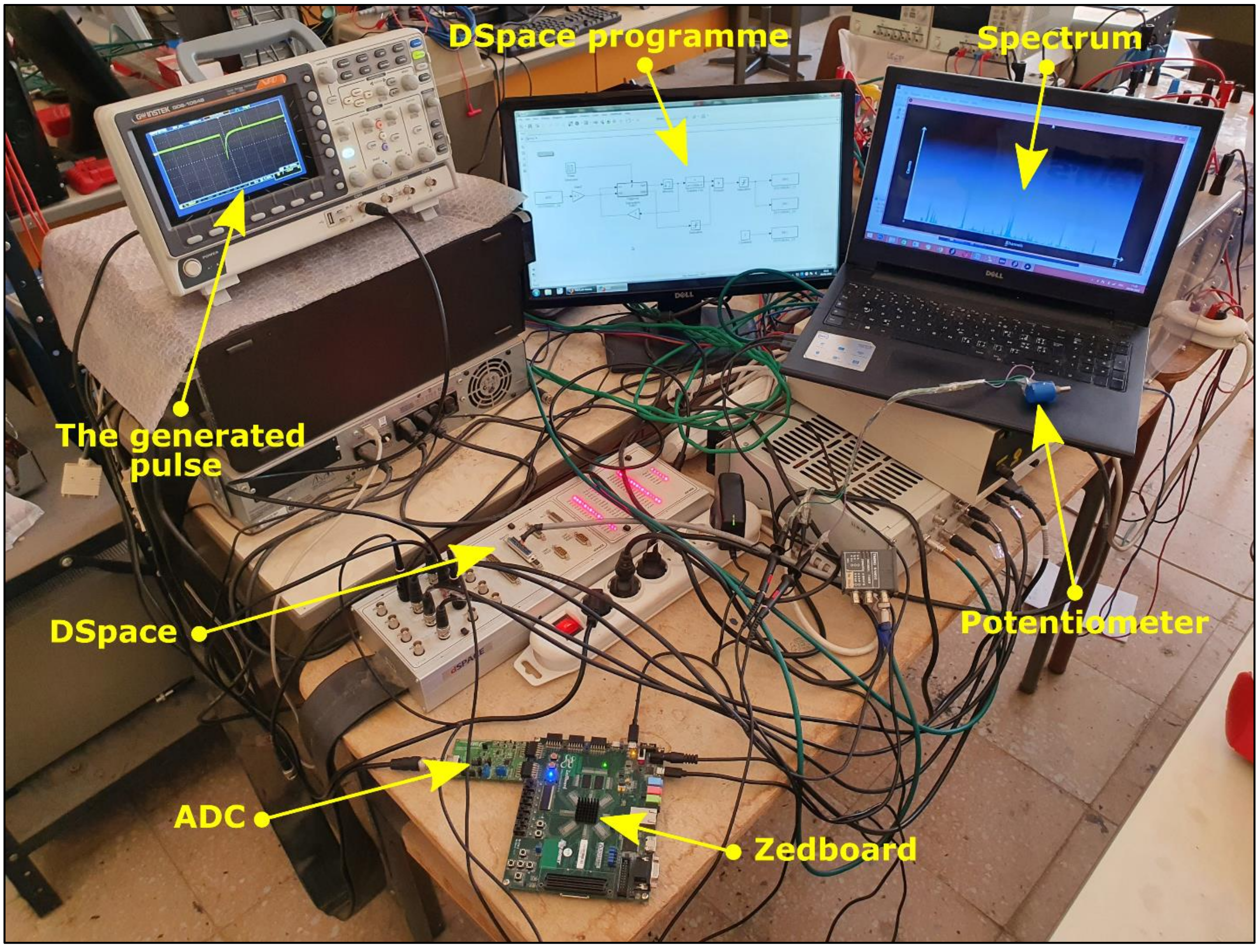


Figure IV.2 Experimental 1st setup.

IV.3 Pulse generating

As discussed before, in reality, the exponential decay pulse signal is the output from the preamplifier preceded by the HPGe detector; However, since access to this sensible part is not allowed in a real gamma ray system, the exponential pulses signal is generated to execute our application using Dspace programmed by MATLAB/Simulink. The scheme of the pulse generating part is shown in Fig IV.3, where the potentiometer used allows us to manually change the amplitude of the pulses, and the Simulink model to program the DSpace is showed in Fig IV.4

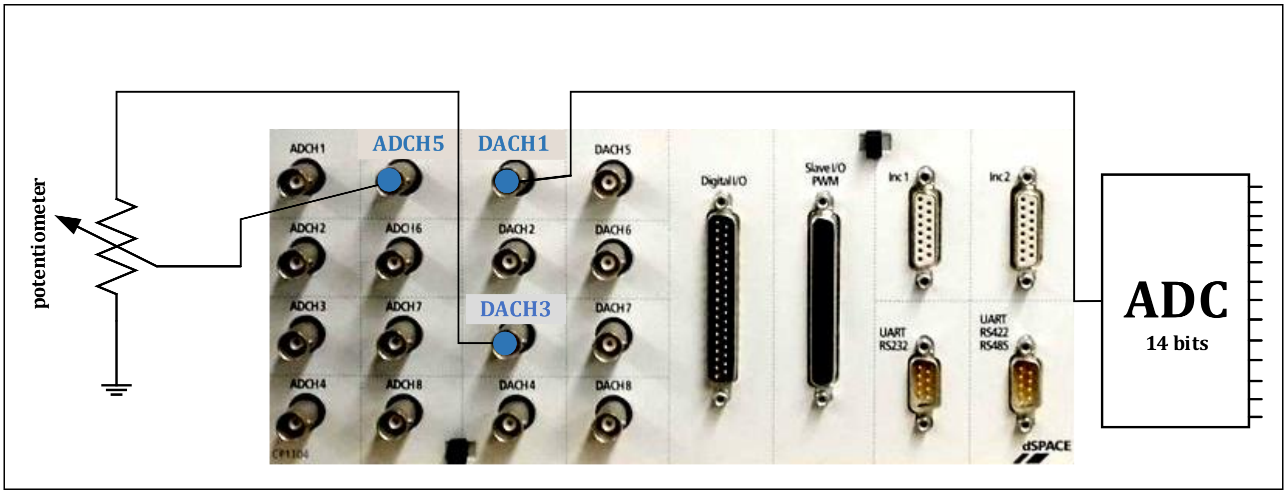


Figure IV.3 The connections of the Dspace.

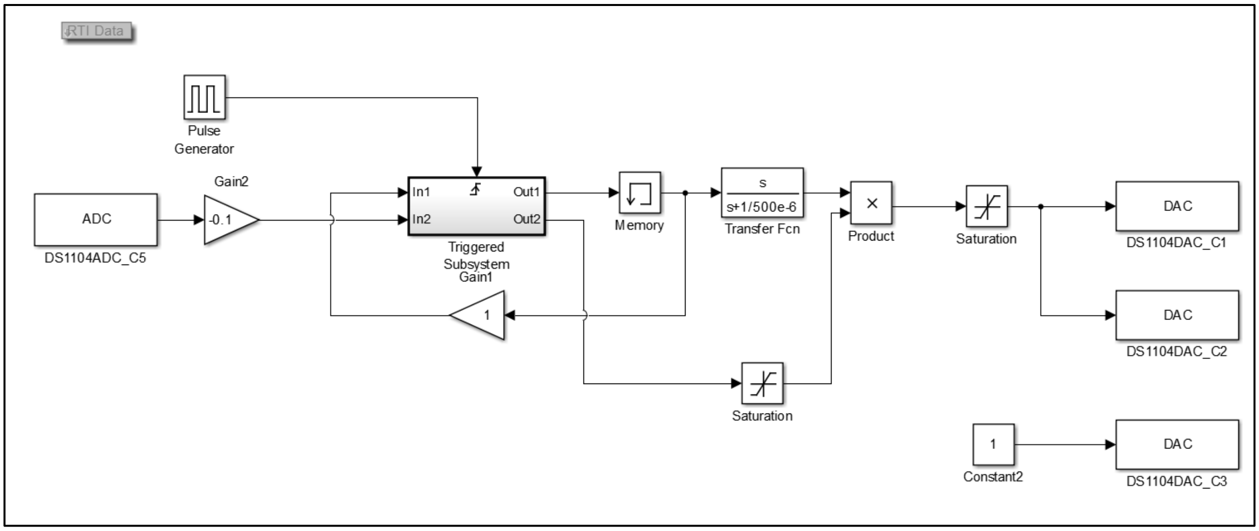


Figure IV.4 The Simulink program of the DSpace

Although, due to some limitations in the Dspace, the following changes have been made:

- The time constant of the exponential decay $0.5ms$
- The period of the success pulses is set to $30ms$
- The sample time in Simulink is fixed to $20\mu s$

Therefore, for the experimental test, the parameters of the system generator blocks were changed to adapt to the pulse generated by Dspace, and as a result of the DAC's resolution of the Dspace, the generated exponential decay pulse have a stairs shape and will cause later stairs drop shape in the flat top of the trapezoidal output resulting a loss in the amplitude of the trapezoidal filter. However, in reality or when using a well-shaped pulse, this problem doesn't exist.

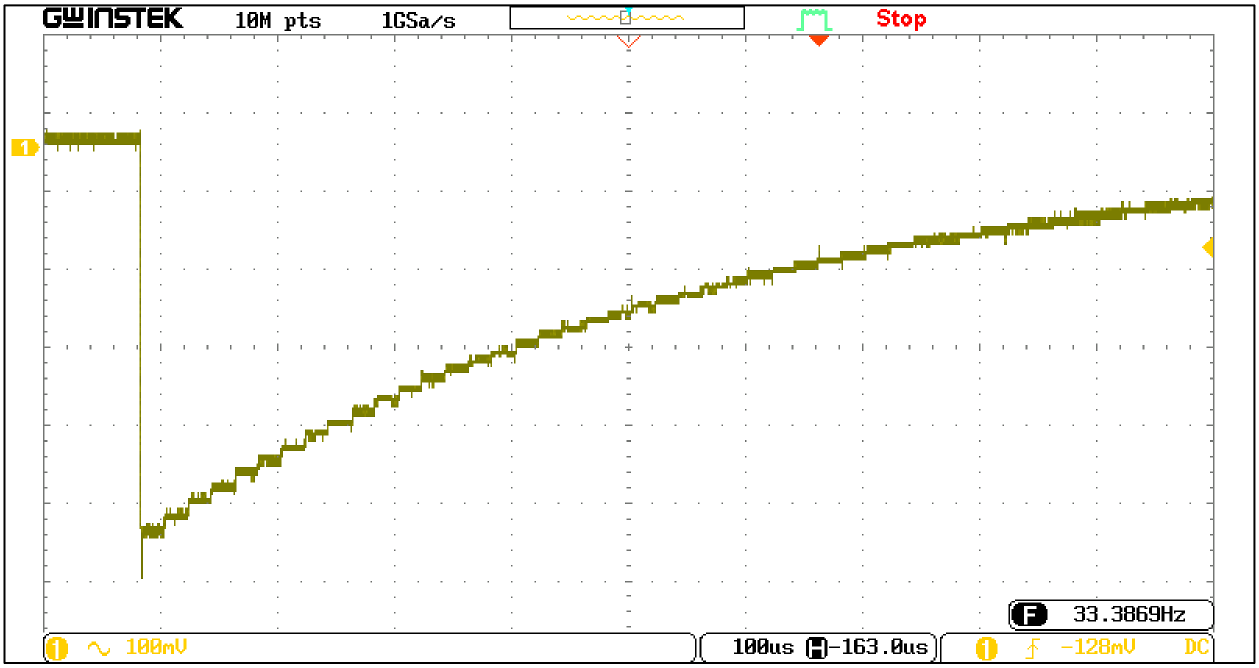


Figure IV.5 The stairs shape of the generated pulse.

IV.4 The ADC board

For the analog to digital conversion of the exponential decay pulse created by DSpace we are using a Pre-Filter ADC board that contain The LTC2248 ADC proceeded by driving circuit and additional circuits in form of stages that perform different functionalities.

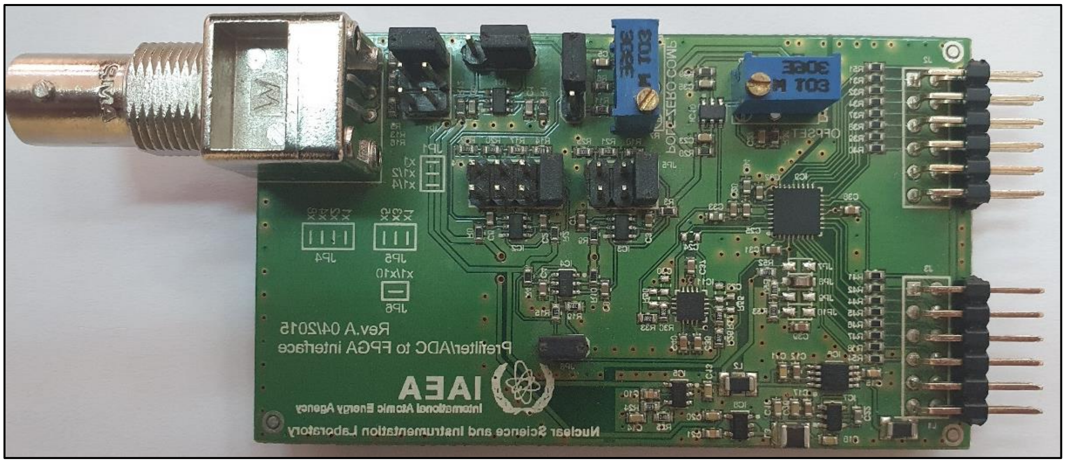


Figure IV.6 ADC board

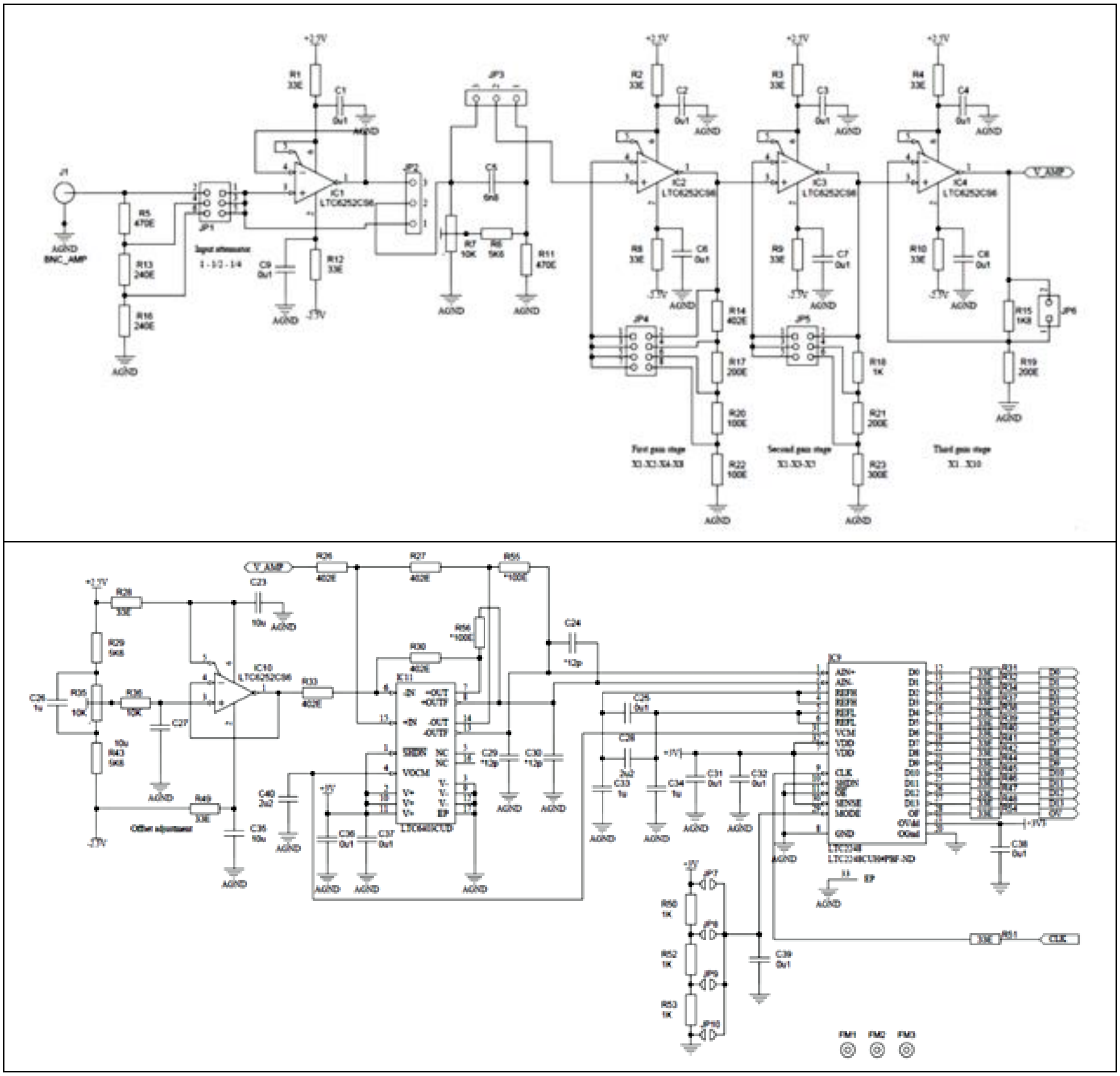


Figure IV.7 ADC board schematic

IV.4.1 Describing the functionality of the ADC board

- The LTC2248 can be used in single-ended ADC mode or in differential ADC mode.
- The Pre-filter ADC board use the LTC2248 ADC IC in differential mode by measuring the voltage difference between two pins (AIN+ - AIN-) which require the use of a differential Input/Output Driver (LTC6403).

- The reason for using differential input mode instead of single-ended input is because of the noise rejection benefit. A differential input better resists electromagnetic interference (EMI) since the EMI which is common in both lines is ignored.
- The reason for choosing the LTC6252CS6 is because it has a good performance combination of high bandwidth, high slew rate, low power consumption and low broadband noise. And they are ideal for lower supply voltage high speed signal conditioning systems.
- The LTC2248's analog differential inputs (AIN+ and AIN-) are driven by the lowpass filtered analog differential pair outputs (+OUTF, and –OUTF) of the LTC6403.The filtered output of the LTC6403 is an Anti-Aliasing Filter, that get rid of the input pulse of higher-frequency noise and signals to ensure that the bandwidth of the signal sampled is limited to the desired frequency range to satisfy the Nyquist-Shannon sampling theorem.
- The LTC6403 op-amp produces differential outputs. By taking a single-ended signal and converting it to fully differential output. The differential output voltage depends on the difference between the two inputs of the op amp.
- The analog differential pair outputs (+OUTF, and –OUTF) depends on the difference between the two inputs of the op-amp (+IN, –IN: Non-Inverting and Inverting Input pins of the amplifier)
- The single-ended signal V_AMP is connected to the LTC6403 by pin +IN via voltage divider while the pin –IN is connected to a variable DC voltage produced by the LTC6252CS6 IC10 and that voltage can be adjusted by the potentiometer R35 to allow controlling the bias (the offset) of the differential output voltages.
- The single-ended signal V_AMP is an image of the pre-amplifier output pulse and it is the result of certain circuit stages that can be controlled by JPs.

IV.4.2 The role of each stage

- IC1 is an isolation amplifier (unity-gain amplifier) that provides isolation between the output of the pre-amplifier (the selected input signal by JP1) and the circuits that are after the IC1.
- The role of the passive filter built around: C5, R6, R7 and R11 is A high pass filter.
- JP3 Allow to choose between a filtered input or a non-filtered input.

IV.5 Zedboard (Zynq™Evaluation and Development board)

To process the pulse and implement the multichannel analyzer we are using the Zedboard which is a low-cost evaluation and development board based on the Xilinx Zynq®-7000 Extensible Processing Platform (EPP) family that is the heart of the board (XC7Z020-CLG484-1). It contains everything necessary to create a Linux, Android, Windows or other OS-based design. The Zedboard incorporates volatile RAM and persistent FLASH memory, buttons, switches, OLED screen, Ethernet connectivity and several interfaces for peripherals and it includes a number of connectors for easy user access to the board's ICs making it an ideal platform for both novice and experienced designers.

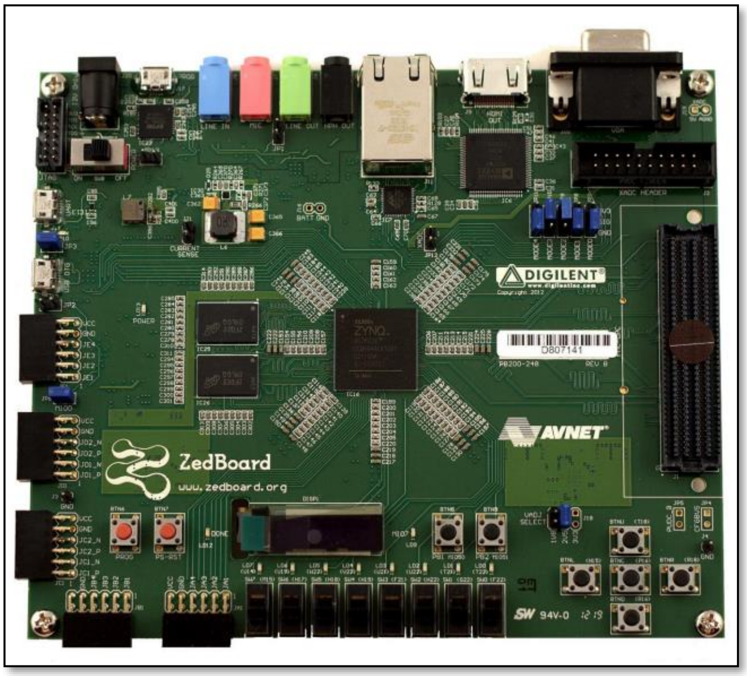


Figure IV.8 The Zedboard.

The main component of the board is the Zynq-7000 System On Chip (SoC) which combines a dual-core ARM Cortex-A9 CPU capable of running full operating systems such as Linux, that is the heart of the Processing System (PS) unit which also includes on-chip memory, external memory interfaces, and a rich set of I/O peripherals coupled with Xilinx Artix-7 FPGA that represents the Programmable Logic (PL) unit along with modules, interfaces and connections devoted to efficient communication between the two allowing to create unique and powerful designs and applications with the Zedboard.

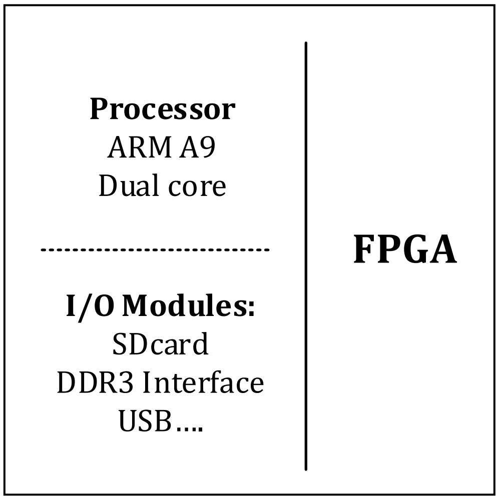


Figure IV.9 The Zynq SoC features a processor and a FPGA as well as I/O peripherals support.

The Zynq SoC can be used in three ways:

- The PS can be used independently of the PL.
- A combination between PS and PL can be used providing an extensible and flexible SoC solution on a single die to achieve complex and efficient designs on the SoC.
- The PL can be designed to operate independently of the PS.

In our Zynq SoC application we combine both PS and PL units of the chip to achieve efficient design on which the PL section that is ideal for implementing high-speed logic, arithmetic and data flow subsystems is used to perform the digital trapezoidal shaping filter, maxima detection, MCU, while the PS that supports software routines and/or operating systems is used to perform the communication between the FPGA and the PC. Using this way, the overall function of the designed system is appropriately partitioned between hardware and software without the overhead of interfacing between two physically separate devices.

The FPGA part is used to perform the digital trapezoidal filtering, measuring the height of the pulse, convert it to channels and then counting the number of peaks that belongs to each channel and store them in the BRAM then send the information to the PS part through AXI4 in form of a 26bits packet (10 bits represent the number of the channel and 16 bits represent the counts at this channel).

In the SDK we program the PS part in C code where we only read the 26 bits from the PL unit that represents the sent packet, to send it again to the PC through USART.

Another application code is written using “processing” software to read the received packet by the COM port from the PS and splits it to extract the channel and its count, then depending on that, displays the histogram which is the spectrum in real-time.

IV.5.1 The links between the PL and PS

The links between the PL and PS are made using standard AXI (Advanced extensible Interface). The Zynq 7000 contains AXI interfaces between the processing system and the programmable logic.

AXI is part of the ARM Advanced Microcontroller Bus Architecture family (AMBA), it is a parallel high-performance, synchronous, high-frequency, multi-initiator, multi-target communication interface, mainly designed for on-chip communication.

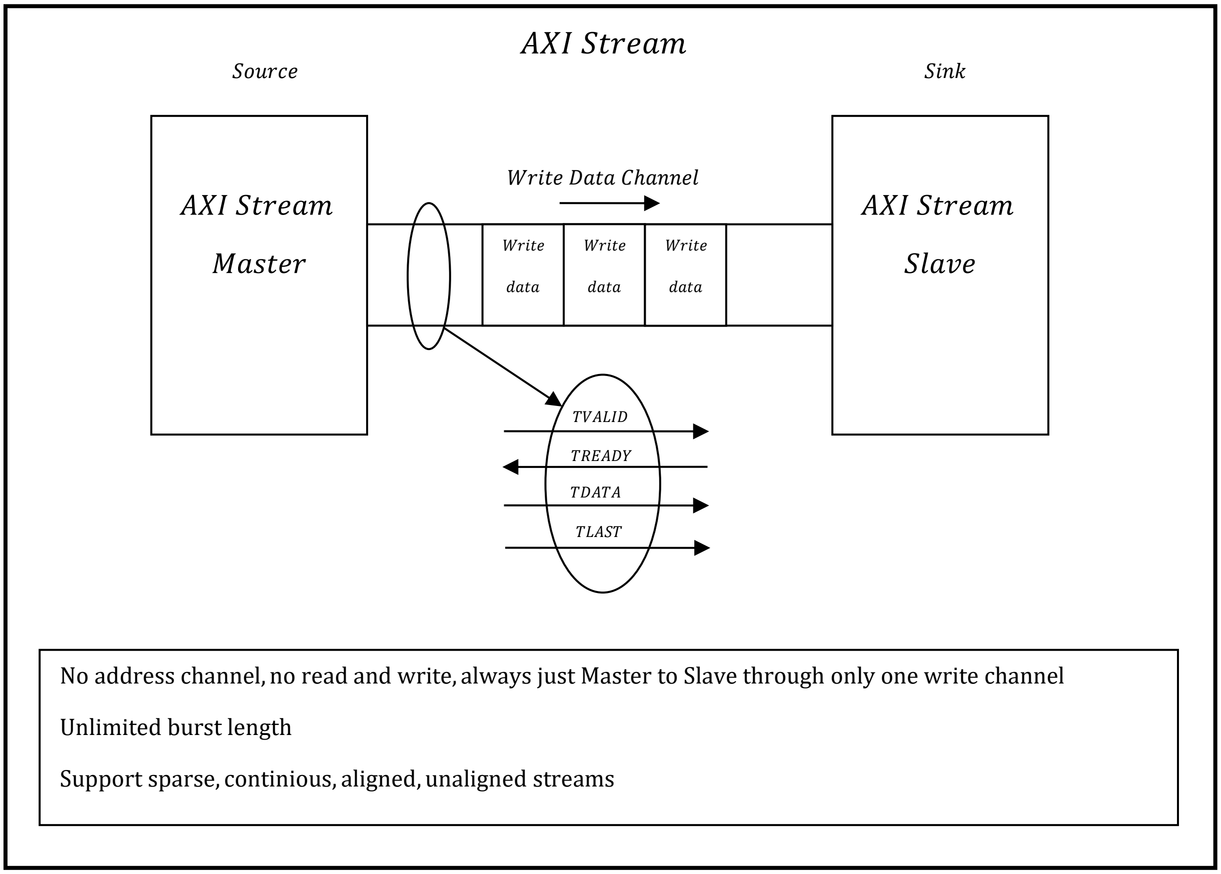
There are three types of AXI4 interfaces:

- AXI4: For high-performance memory-mapped requirements.
- AXI4-Lite: For simple, low-throughput memory-mapped communication (for example, to and from control and status registers).
- AXI4-Lite is similar to AXI4 with some exceptions: The most notable exception is that bursting is not supported; it allows only one data transfer per transaction.
- AXI4-Stream: For high-speed streaming data.

Both AXI4 and AXI4-Lite interfaces consist of five different channels: Read Address Channel, Write Address Channel, Read Data Channel, Write Data Channel and Write Response Channel, each channel consists of a number of signals that performs a specific operation of the channel. The data can move in both directions between master and slave simultaneously, and data transfer sizes can vary; however, The AXI4-Stream protocol defines a single unidirectional channel for transmission of streaming data. The AXI4-Stream channel models the write data channel of AXI4. Unlike AXI4, AXI4-Stream interfaces can burst an unlimited amount of data.

Because that our application typically focuses on data-flow, where the concept of an address is not present, the AXI4-Stream interface is used to transfer the 26bits data from PL (The AXI-Stream Master) to PS (The AXI-Stream slave). Note that in the Zynq-7000 SoC devices The PL-based cores use AXI4 or AXI4-Lite while the processing system block uses AXI3 or AXI3-Lite interfaces (AXI3 is a subset of AXI4). Therefore, a conversion is required, typically through an

AXI interconnect core and lucky for us the Xilinx tools automatically insert the necessary adaptation logic to translate between AXI3 and AXI4.



No address channel, no read and write, always just Master to Slave through only one write channel

Unlimited burst length

Support sparse, continious, aligned, unaligned streams

Figure IV.10 AXI Stream architecture.

The stream protocol minimizes overhead by removing the need for addressing keeping only one channel (Write data Channel) that consist of 4 signals:

- **TVALID:** Bus signals indicate when data is available,
- **TREADY:** Receiver can optionally specify ready,
- **TDATA:** Where the Data is sent,
- **TLAST:** Signal end of packet of data.

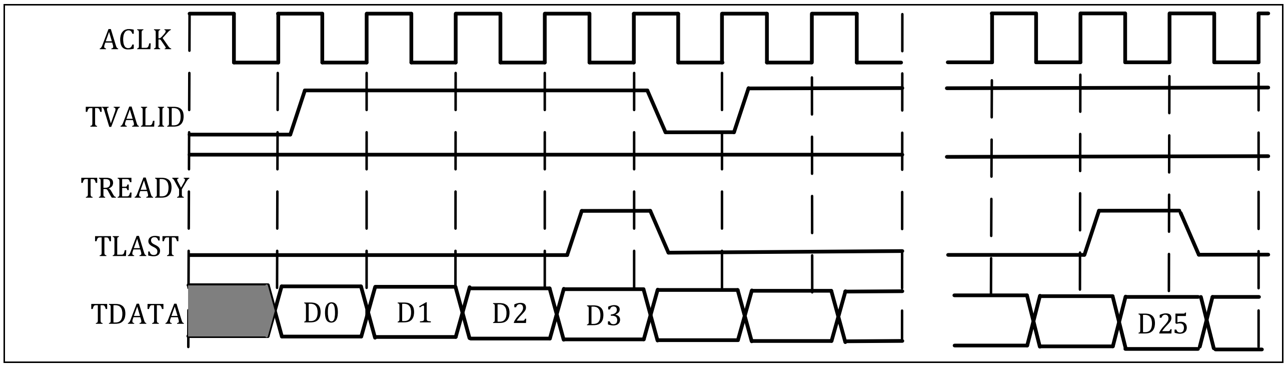


Figure IV.11 The write data channel of the AXI Stream.

Other signals might be available as shown in the table, which are optional and required for more advanced applications.

Table IV.1 AXI4 signals and their status

Signal	Status	Notes
TVALID	Required	
TREADY	Optional	TREADY is optional, but highly recommended.
TDATA	Optional	
TSTRB	Optional	Not typically used by endpoint IP; available for sparse stream signaling. **Note:** For marking packet remainders; TKEEP use rather than TSTRB
TKEEP	Optional	Null bytes are only used for signaling packet remainders, leading or intermediate Null bytes are generally not supported.
TLAST	Optional	
TID	Optional	Not typically used by endpoint IP; available for use by infrastructure IP.
TDEST	Optional	Not typically used by endpoint IP; available for use by infrastructure IP.
TUSER	Optional	

The Xilinx software hides all the above and other complex details and simplify the use of the AXI interfaces. (The mentioned information above was collected from the Vivado Design Suite: AXI Reference Guide[25]

IV.5.2 Programming the Zedboard

Programming the Zynq for our application involves two tasks: designing the digital circuit for the FPGA and programming the processor. These two tasks can be done nearly independently by different tools:

- The FPGA design can be done by using the Vivado Block Design (BD), which is a top-level schematic that connects different blocks. The blocks contain the main functionality and can be designed in different ways. One is using a hardware description language like VHDL or Verilog. The other option is to rely on high level synthesis, like Vivado HLS or MATLAB based tools (System Generator, HDL

Coder)[26] that basically try to convert the Simulink model into a digital circuit which is our case. Note that using MATLAB to design our PL part allowed us to clearly analyze our design and to optimize it easily as shown in the simulation section.

- The ARM processor can be programmed either directly in C/C++ by the Xilinx SDK tool for example, or using an operating system that would be installed on the SD Card. But we are going to program it directly in C/C++ using the Xilinx SDK software after building our application’s hardware.

IV.5.3 General Design Flow

The general design flow of our application (spectroscopy system) involves three phases of coding:

1. Programming the FPGA in Vivado.
2. Programming the processor in SDK.
3. Programming software in PC using Processing.

IV.5.3.1 Vivado:

To program the FPGA in Vivado we follow these steps:

- Open Vivado and select Zedboard
- Create a new Vivado Project
- Create empty block design workspace inside the new project
- Generating the created IP by MATLAB that contain the implementation of the trapezoidal filter, maxima detection and the multichannel analyzer and adding it to Vivado.
    a. In the Vivado project, click on "IP Catalog". In the IP Catalog window, right click on any empty area, select "IP Settings...".
    b. In the pop-up dialog, press on "Add Repository..." button from "Repository Manager" and provide the path to the IP location (<Target Directory>/Ip) and click OK.
    c. Find the IP category from IP Catalog, IP will be under this category. Double click to add the IP to the project.
    d. Under the IP category from IP Catalog, IP will be under this category. Double click to add the IP to the project.

- Add required IP blocks using the IP integrator tool and build Hardware Design and configure the clock wizard block to generate the clocks of both FPGA and the ADC as shown in Fig IV.12
- Validate and save block design
- Create HDL system wrapper
- Define the physical constrain file (the .xdc file) using TCL coding language, where we describe all the used pins.
  a. Define the used pins package.
  b. Define I/O Standard of the used pins to be LVCMOS 3.3v.
  c. Add a Pullup to the input data pins of the ADC.
  d. Set the Output Drive Current (Drive strength) of the adc_clock pin.
  e. Set the Output Slew Rate of the adc_clock pin.
- Run design Synthesis and Implementation
- Generate Bitstream File
- Export Hardware Design including the generated bit stream file to SDK tool

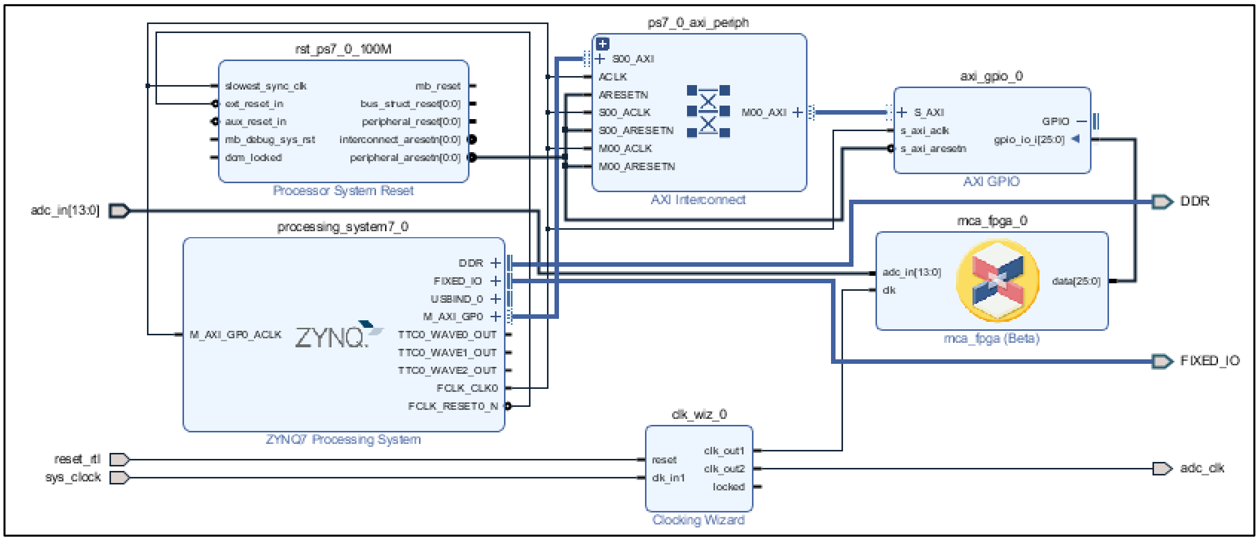


Figure IV.12 Vivado Block design for our spectroscopy system

Now the Hardware design is exported to the SDK tool. The Vivado to SDK hand-off is done internally through Vivado.

IV.5.3.2 SDK:

We use SDK to create a Software that will use the customized board interface data and FPGA hardware configuration by importing the hardware design information from Vivado.

- Launch SDK
- Create new application project and write the C code for the PS.
- Program the FPGA from the SDK and run the application on the processor.

IV.5.3.3 Processing:

A software called Processing is used to display the spectrum in the PC's screen after reading the data from the COM port. It's a flexible software and language sketchbook, for learning how to code in the context of the visual arts [27].

IV.6 1st setup result

The spectrum shown in Fig IV.13 is obtained after passing through all the steps mentioned earlier where we changed the amplitude of the pulses during time using a potentiometer.

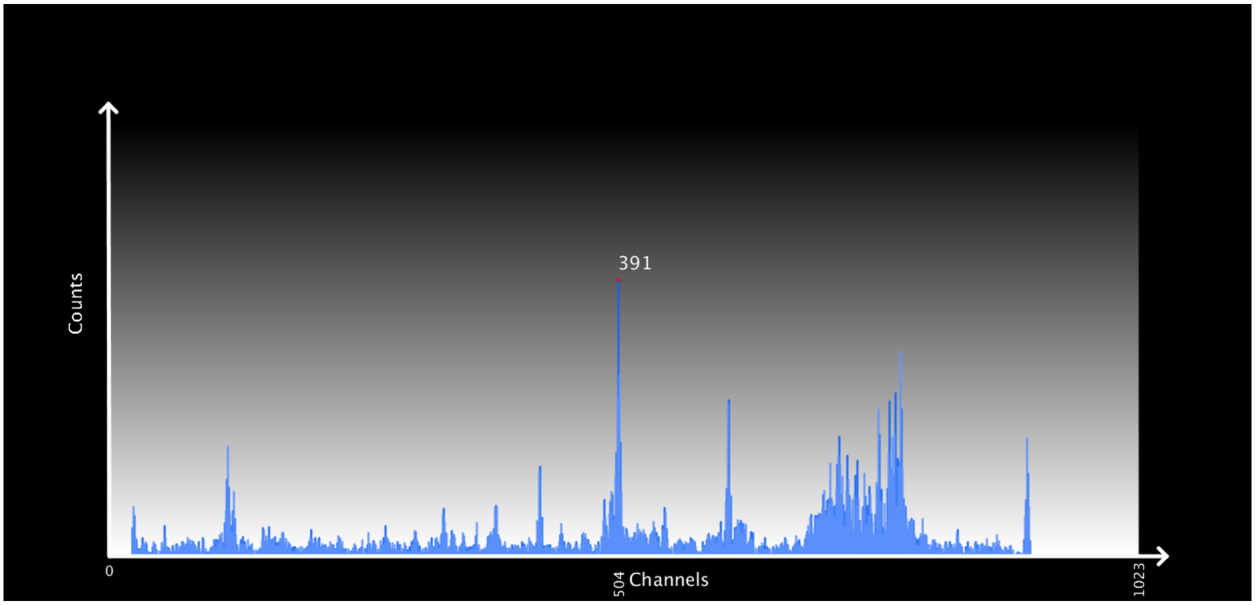


Figure IV.13 The resulted spectrum for a random pulse height controlled by the potentiometer.

We can see a histogram that shows the number of counts in function of the channels which is the spectrum in real-time, the red dot and the number above it is the highest number of counts and the number in the channel axis is its channel number. Generally, the number of channels is set to 4096 but due to the limited pixels of our screen it was reduced to 1024 channel. To validate our work, we tested that the counting is performed correctly without any pulses loss and the channel are calculated correctly for both small and high amplitudes pulses.

IV.6.1 The channels test

To test the channels, we sent a periodic sequence of negative exponential decay pulses with a fixed amplitude, and visualize the trapezoidal output of one pulse to obtain the spectrum.

In order to visualize the trapezoidal output, we designed another programmable logic design in Simulink that store the result of a unique pulse after detecting it in the RAM and then send it to the PC to visualize it.

As discussed before, using DSpace to generate the pulse signal leads to a loss in the calculated pulse height due to the stairs shape of the generated pulse which is resulted in the tests.

IV.6.1.1 For small amplitude

An amplitude of -0.21 volt was fixed for the pulses and we visualize the trapezoidal output and its spectrum as shown in Fig IV.15 and Fig IV16

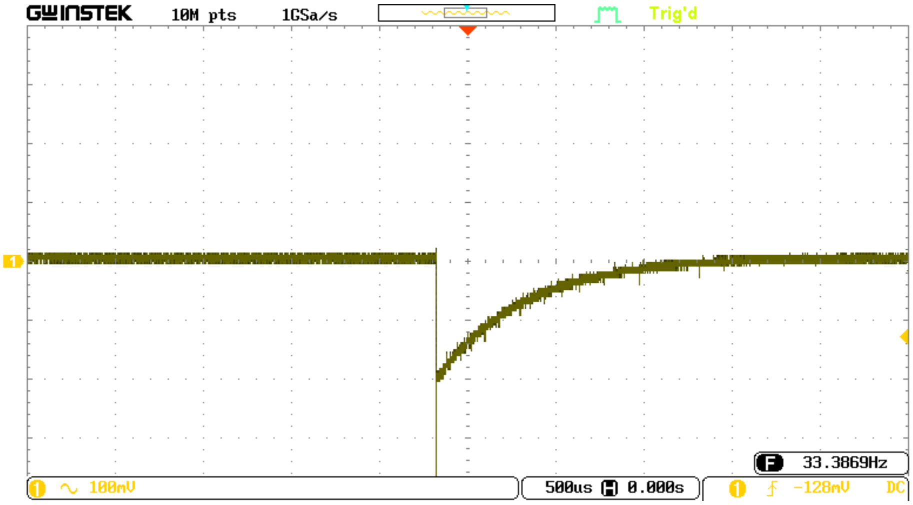


Figure IV.14 Oscilloscope view of the generated small pulse.

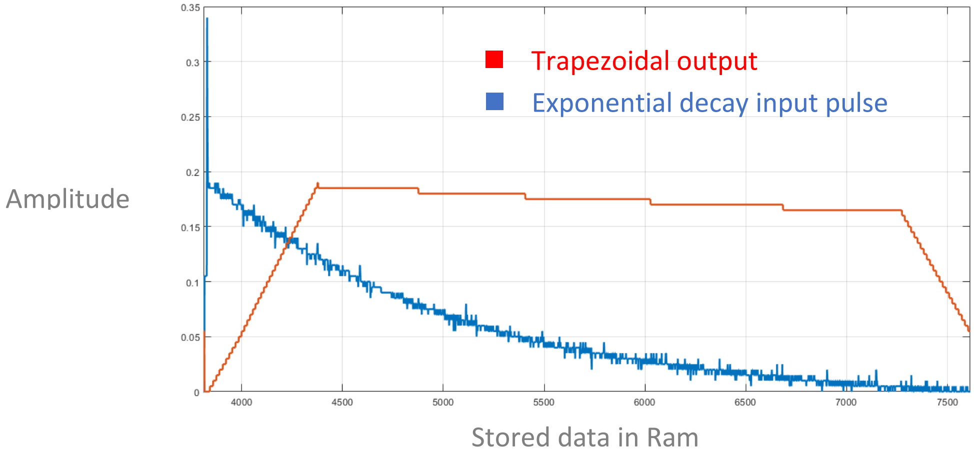


Figure IV.15 The recorded input pulse by the FPGA and the trapezoidal output for the small amplitude test.

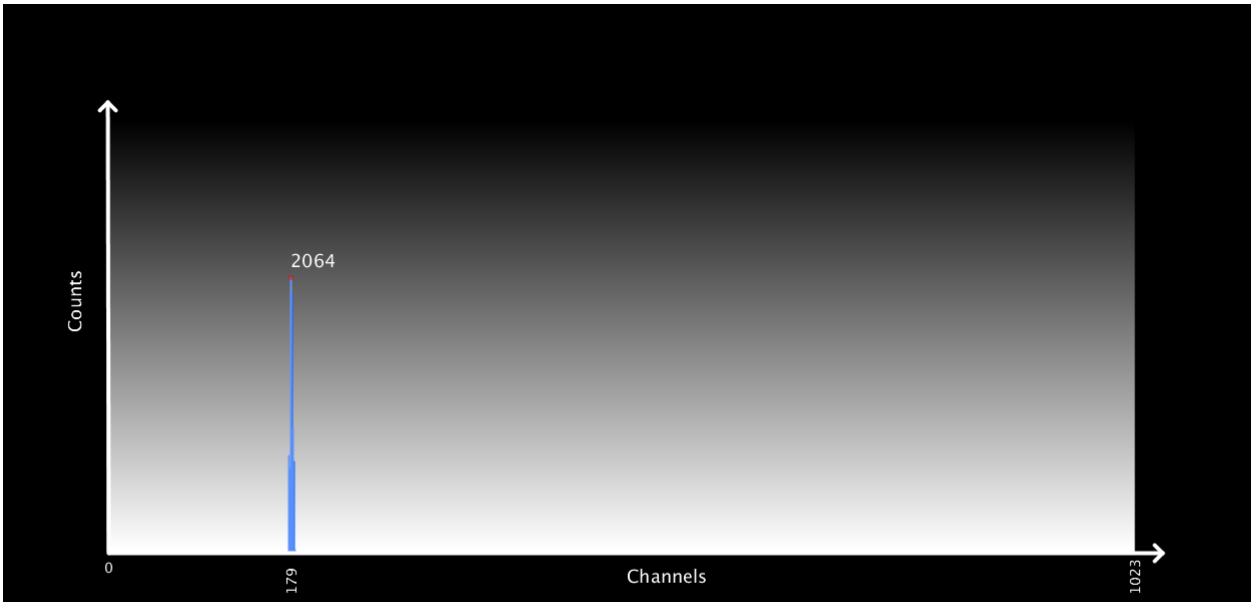


Figure IV.16 The resulted spectrum of a fixed -0.21v amplitude successive pulses for some duration

The resulted channel is supposed to be 214 but because of the small loss in the trapezoidal output, it results a fundamental 179 channel.

IV.6.1.2 For High amplitude

With the same test but with different amplitude of -0.97 volt we obtain the following:

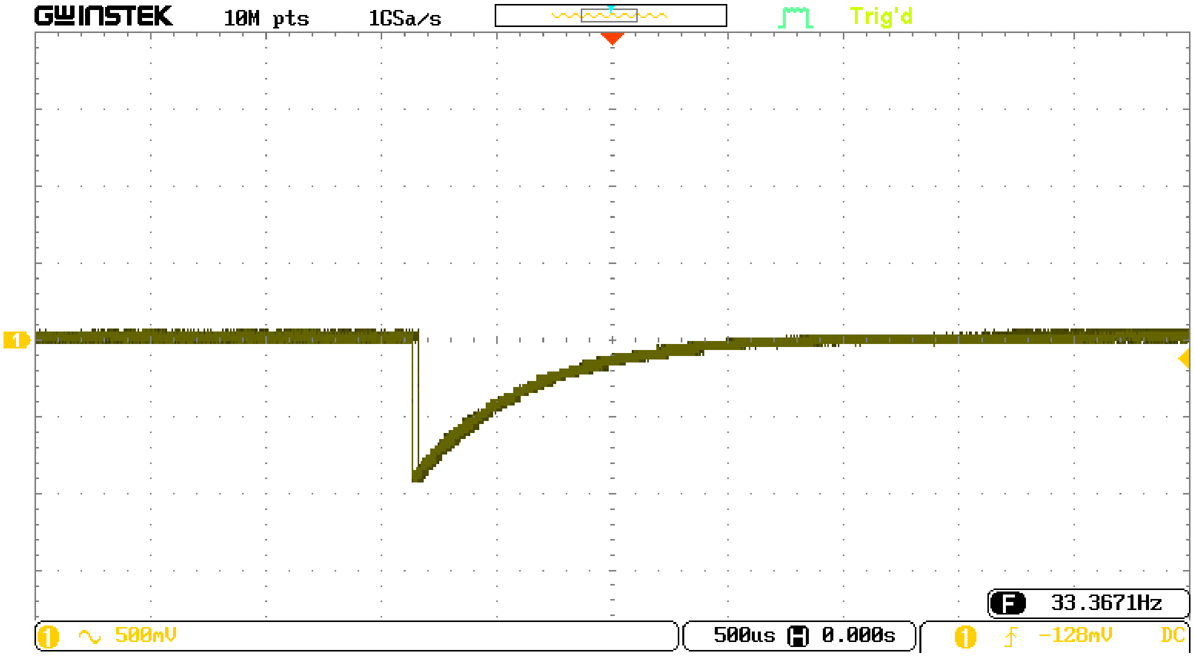


Figure IV.17 Oscilloscope view of the generated high pulse.

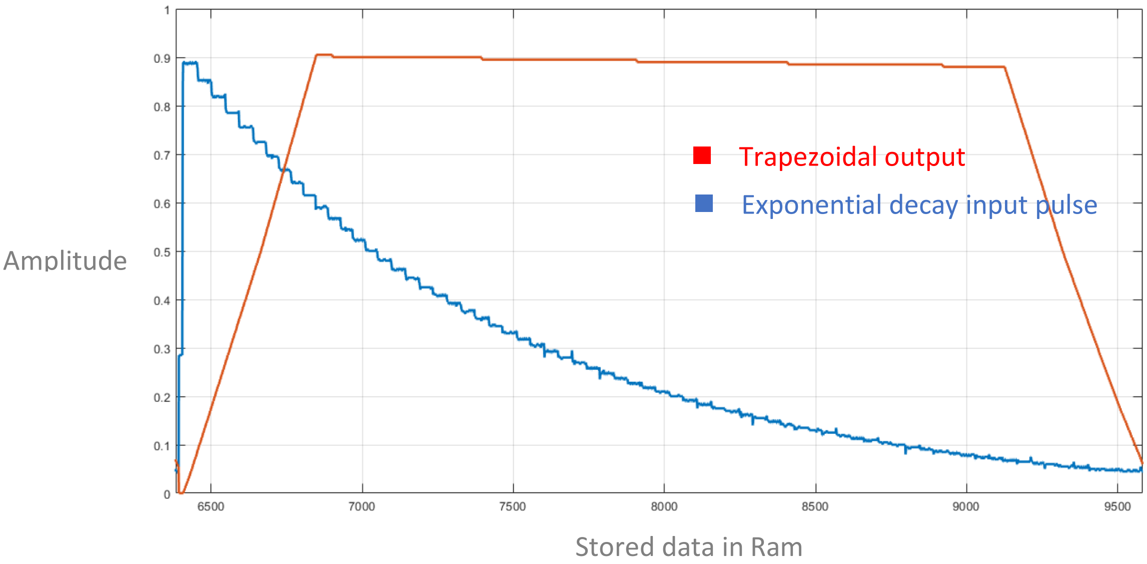


Figure IV.18 The recorded input pulse by the FPGA and the trapezoidal output for the high amplitude test.

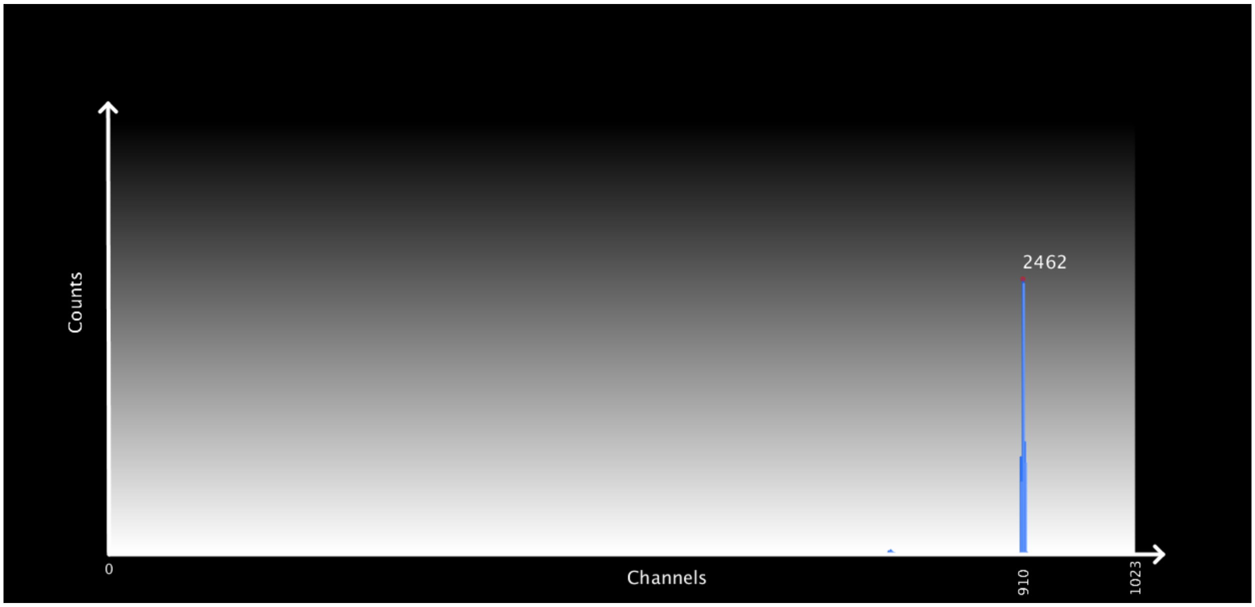


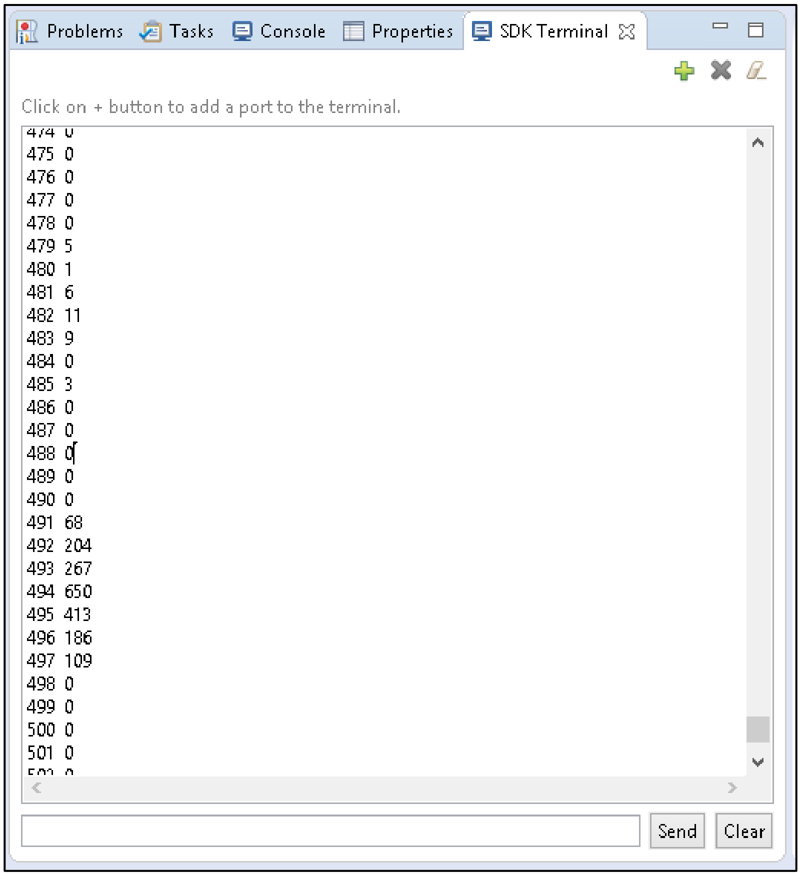
Figure IV.19 The resulted spectrum of a fixed -0.97v amplitude successive pulses for some duration.

The same as the test with the small amplitude the result of the spectrum for the high amplitude also have a loss in the detected channel due to the reason described before.

IV.6.2 The counts test

For the count test we send a series of pulses for a specific duration then we sum the number of counts in the entire channels and we compare it with the expected counts for that specific duration.

- Fig IV.20 shows the corresponding counts for each channel after recording a series of pulses with a fixed -0.5v amplitude for approximately 1min duration. since the period of the success pulses is set to $30ms$, the total counts for 1min suppose to be 2000 counts; however, the total number of counts from the figure is 1932 which is less than 2000 and this is only because of not precisely set the1min test and it have no relation with any count loss.



```
474 0
475 0
476 0
477 0
478 0
479 5
480 1
481 6
482 11
483 9
484 0
485 3
486 0
487 0
488 0
489 0
490 0
491 68
492 204
493 267
494 650
495 413
496 186
497 109
498 0
499 0
500 0
501 0
502 0
```

Figure IV.20 The corresponding counts for each channel for the count test in the serial monitor of the SDK.

IV.7 Experimental 2nd setup

A second setup that gives better results was used as shown in Fig IV.21 where the exponential decay pulse in this setup was generated using a programmable function generator instead of the DSpace and that gives us the advantage to freely control the parameters of the pulse and gives better resolution.

A “.csv” file was generated by MATLAB that include both the time and the amplitude values of the exponential decay pulse and it was regenerated using “EasyWave” software of the used function generator then passed through a USB device to the function generator.

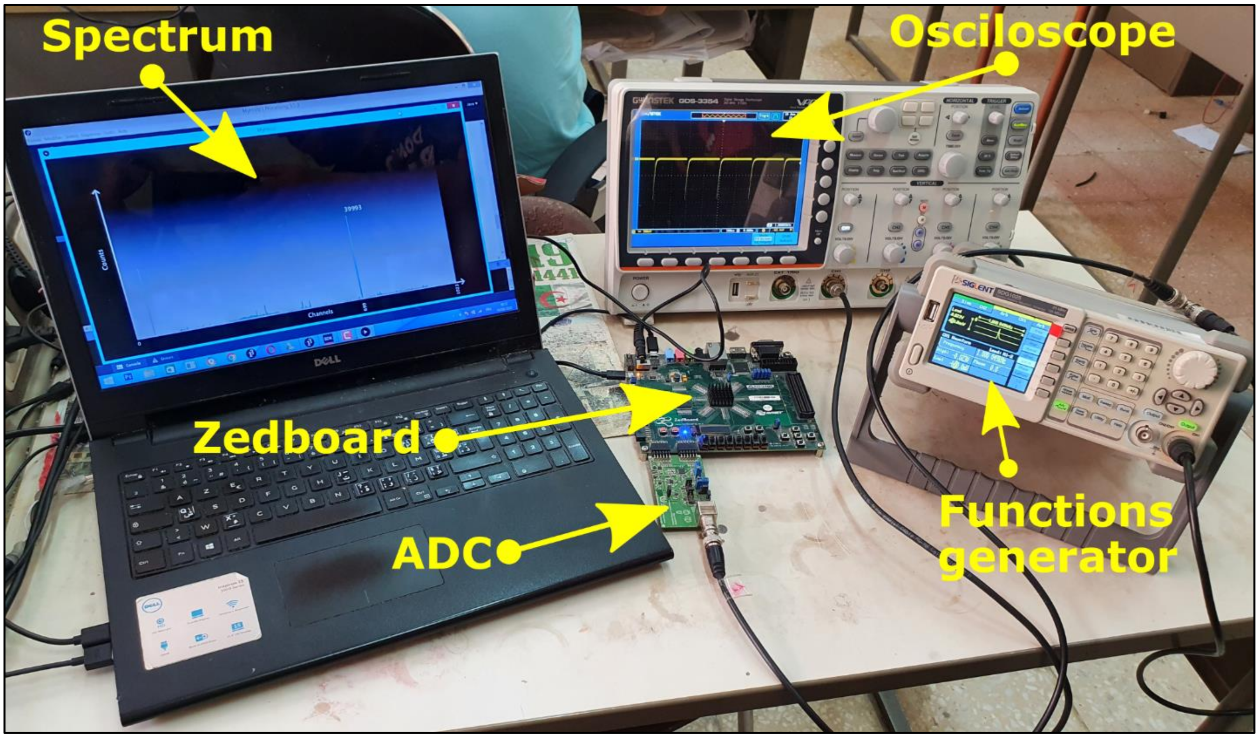
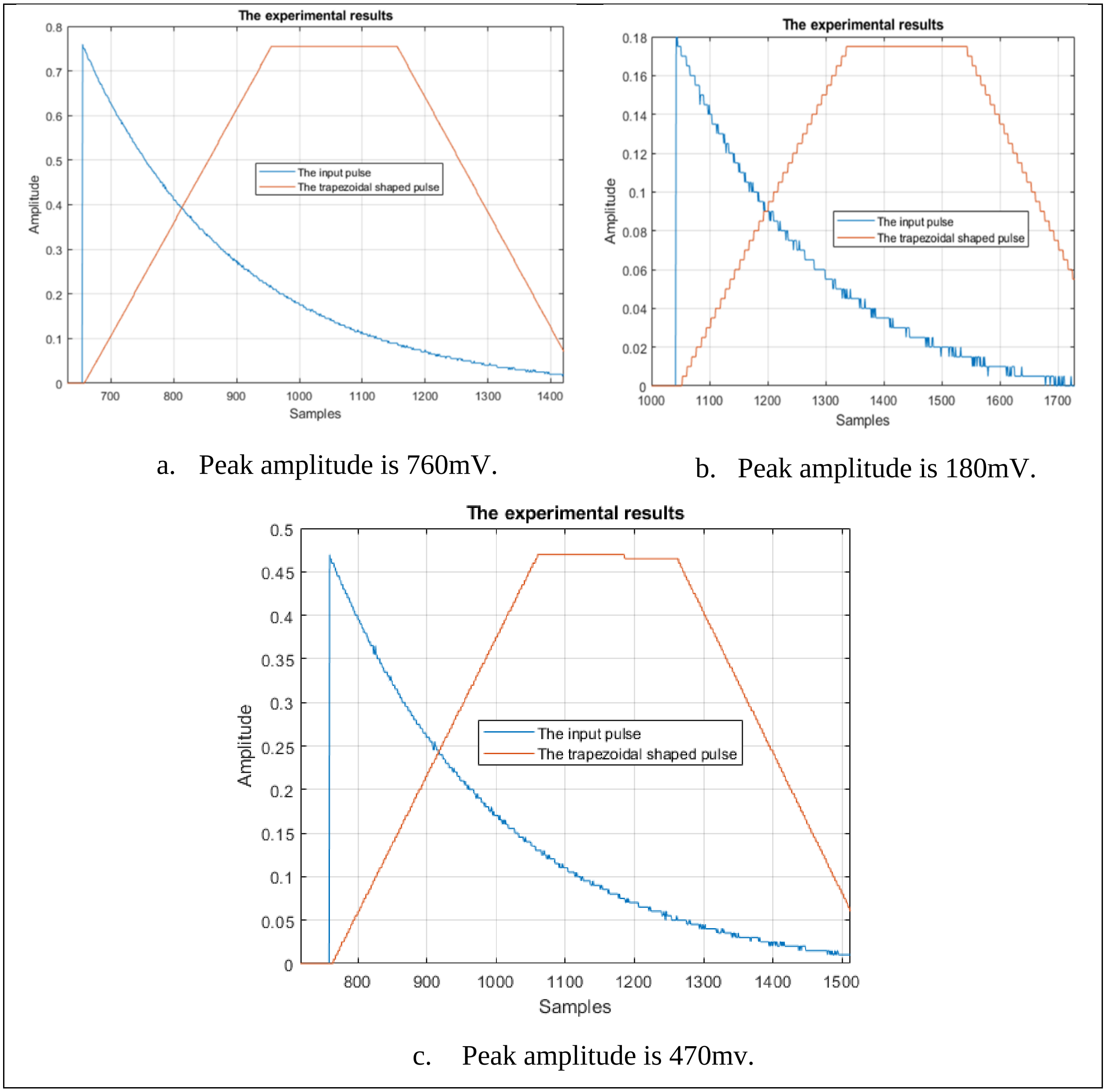


Figure IV.21 Experimental 2nd setup

IV.7.1 The 2nd setup results

By using the same technique used as the 1st setup, the input and the output of the DTS filter, we recorded the following results for large, medium and small amplitude.



a. Peak amplitude is 760mV.

b. Peak amplitude is 180mV.

c. Peak amplitude is 470mv.

Figure IV.22 The recorded exponential input pulse and the DTS output for different peaks.

The shape of the resulted trapezoidal shape is pretty good; however, few things have to be pointed. The X-axis is limited before the end of the trapezoidal output and that is because we are storing the data that is above a threshold value so the hidden values were not recorded as they were under the threshold that was used to detect the pulse and enable the storing in the RAM. Also, in Fig IV.22.c for we view that there is distortion at the end of the flat top region of the trapezoidal shape and that is due to the small changes in the time constant parameter T_d that was noticed during the change of the different amplitude in the pulses.

To obtain the spectrum in real time we will do as the experimental 1st setup; therefore, we recorded the following results for large, medium, small and random amplitudes as in the Fig IV.23, Fig.24, Fig.25, Fig.26

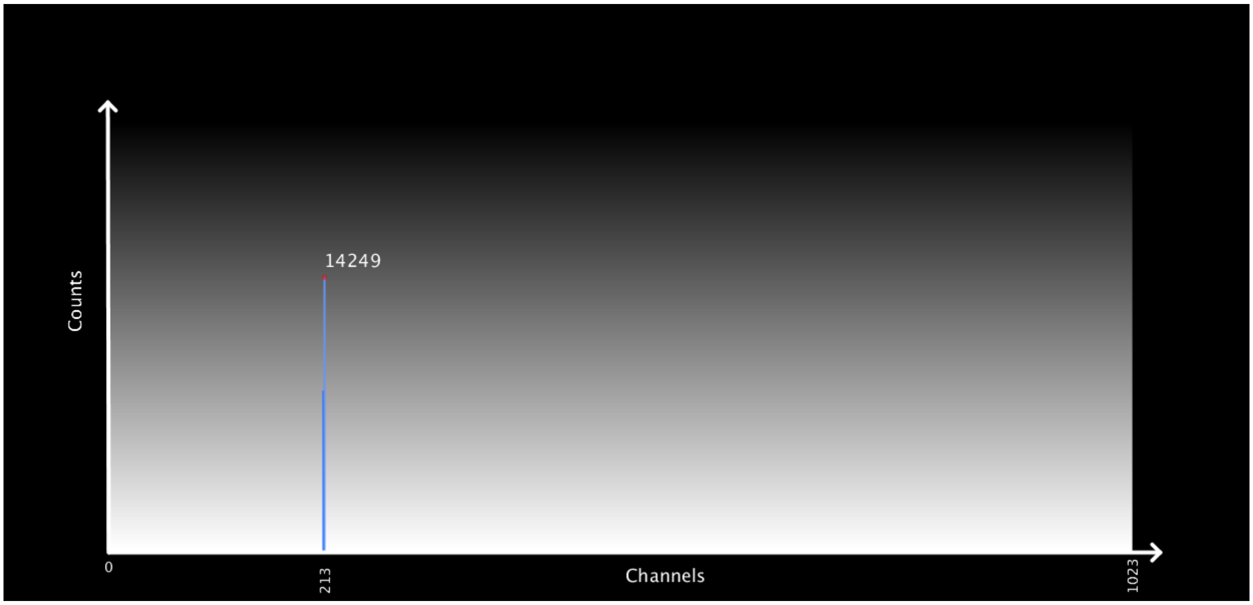


Figure IV.23 The resulted spectrum of a fixed -200mv amplitude successive pulses for some duration

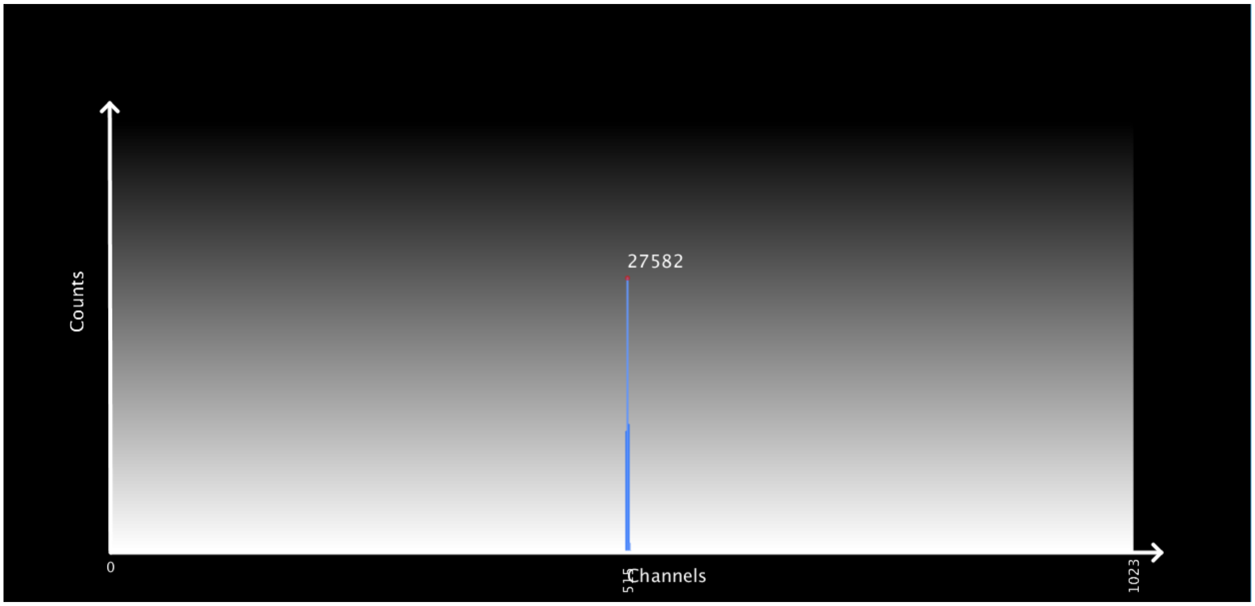


Figure IV.24 The resulted spectrum of a fixed -500mv amplitude successive pulses for some duration

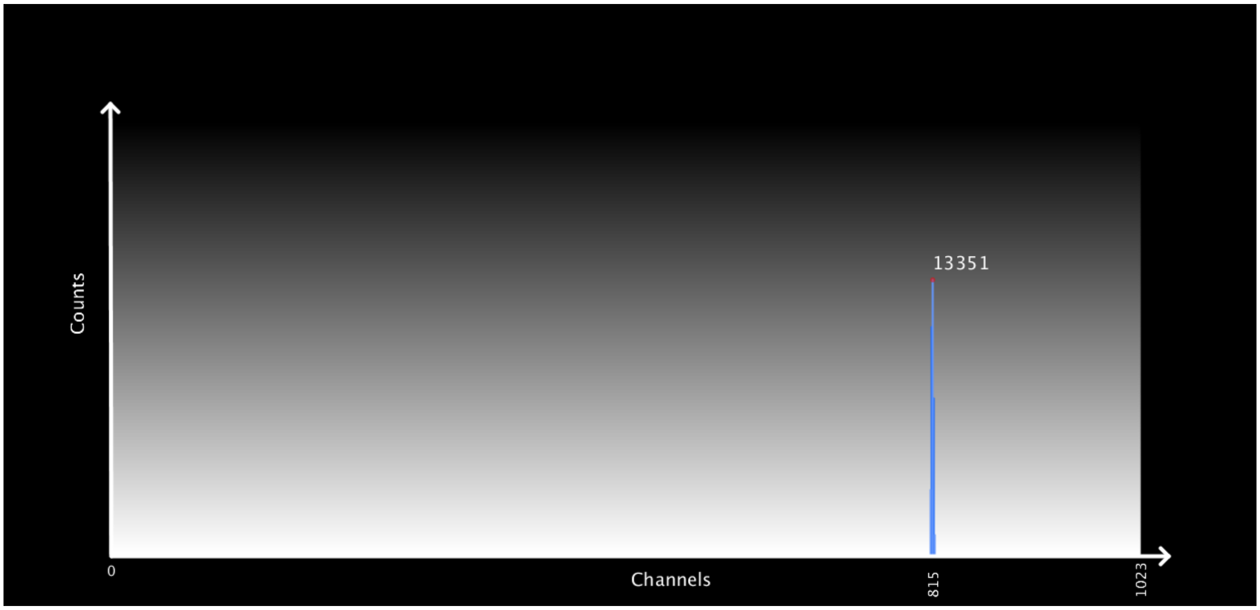


Figure IV.25 The resulted spectrum of a fixed -800mv amplitude successive pulses for some duration

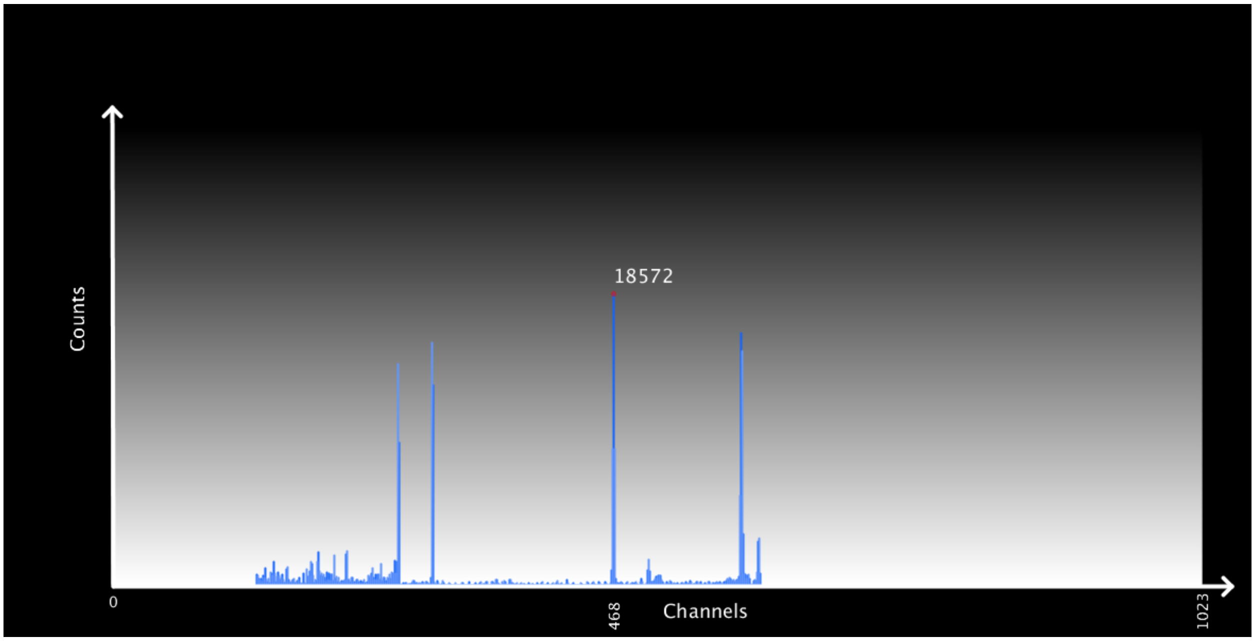


Figure IV.26 The resulted spectrum for a random pulse height controlled by the function generator.

As shown in the previous figures the channels were detected more precisely than the ones in 1st setup due to the appropriate shape of the trapezoidal output caused by the nice shape of the generated input pulse by the function generator.

IV.8 Conclusion

The results shown in this chapter validate the designed work that combines different hardware components and software parts in order to perform different tasks that works together to display spectrum in real time. The obtained results in this chapter from the 2nd setup are better than the1st setup and that shows the advantage of using the function generator that gives the ability to generate a series of exponential pulse with smaller T_d, higher frequency, lesser noise and smoother shape.

General conclusion

In first, it was required to understand the physics behind the gamma rays in order to clearly understand the interaction of those photons with the materials of the detector, to create the exponential decay pulse at the output of the preamplifier that it height is generally proportional to the energy of the incident photon; therefore, by measuring the peaking value of the pulses and classifying them we can represent the number of the incident photons for each energy level.

It was also required to understand the different phenomenon in the spectroscopy chain as ballistic deficit, pulses pile ups in high radioactivity and different noises that effect the resolution of the system and cause an error in the measurement of the energy. In order to design an efficient spectrometer, we need to take into account these problems to increase the resolution and the throughputs of the processed pulses.

As the digital processing of the pulses including the digital shaping of the pulse, detecting the maxima and classifying the height of the pulses requires a high speed and a significant hardware resource, it was implemented on an FPGA while the other tasks was handled by another software. Combining the hardware and the software this way allows to achieve an efficient design; moreover, an analysis of the arithmetic operations errors in the FPGA was performed in order to optimally design this arithmetic operation.

We are satisfied with the obtained results; however, the next step in the project is to test it with a real pulse that come from a preamplifier proceeded by HPGe detector. And future work can be done to this project including using the S-K filter, pull up rejection, baseline restoration, pulse detection and more other options to increase the performance of the system and therefore the resolution of the spectrum. After the spectrum display phase is done it is also possible to perform a spectrum analysis to the resulted spectrum in order to detect nuclides. We can also decrease the spectrum display duration by using interrupts and send only the new values of the counts.

Appendix

MATLAB Code

```
close all
clear
clc
%% parameters

%simulation parameters
simulation_clock = 1e-9;
simulation_start_time = 0;
simulation_end_time = 6e-4;

%preamplifier model parameters
Td = 5e-6; %high pass filter differentiation constant (decay time constant
of the preamplifier pulse)
Tpprd = 100e-6;  %pulse period

%ADC parameters
ADC_resolution = 14; %ADC resolution
B_ADC = ADC_resolution-1; %number of binary places to the right of the
binary point.
ADC_clock = 2e-8; %ADC sampling time for 50Ms/s

%FPGA parameters
delta_t = 10e-9; %sampling time (clock period of the FPGA)

%DTS filter parameters
t1 = 3e-6;    %desired rising time of the trapezoidal signal
t2_t1 = 2e-6; %desired flat top time of the trapezoidal signal

beta = exp(-delta_t/Td); %the DTS filter coefficient beta

A = fix(t1/delta_t);
A_delay = A-3;

B = fix(t2_t1/delta_t);
AplusB_delay = A+B-3;

%derivative parameters
c = 1e-4;
a = 0.001;
```

The Mcode of the sign inverter

```
% Input: x : Signed number
% en: bool:
% Output: y
% calculates:
% y = -x if en = true
% y =  x if en = false
% MCode block literature
% Title: System Generator for DSP Reference Guide, UG638,
% File name: sysgen_ref.pdf, Chapter 1
% Edition: v11.4 December 2, 2009, page 220 - 241
function y = sign_inverter(x,en)
%output result depending on 'en' port
if(en)
    %reinterpret xsgninvbin with same format as x
    %extract data type properties of the input number x
    arith = xl_arith(x);
    if(arith ~= xlSigned)
        error('my message input must be signed number');
    end
    nbits = xl_nbits(x);
    binpt = xl_binpt(x);
    %in 2's complement sign inversion is done by the following procedure:
    %1.  number is reinterpret as unsigned integer
    xunsigned = xl_force(x,xlUnsigned,0);
    %2. each bit is inverted (xl_not) and 1 is added
    xsgninvbin = xfix({xlUnsigned,nbits,0},xl_not(xunsigned)+1);
    %3. result is reinterpret in original representation
    xsgninvxfix = xl_force(xsgninvbin,arith,binpt);
    disp(['nbits=',num2str(nbits),' binpt=',num2str(binpt),' x
Unsigned=',num2str(xsgninvbin)]);
    y = xsgninvxfix;
  else
    y = x;
end
```

The Mcode used in part 3 in the MCU subsystem

```
function maxima1 = xlmax(max_detec_clk, prev_max_detec_clk, input1)
  if prev_max_detec_clk == 0

    if max_detec_clk == 1

        if input1 > 0
        maxima1 = input1;
        else
        maxima1 = 0;
        end


    else
        maxima1 = 0;
    end

  else
    maxima1 = 0;
  end
```

References

[1] "Class - 12-Physics - Alternating Current and Electromagnetic Waves - PPT-3 of 3 | PDF | Electromagnetic Spectrum | Electromagnetic Radiation," *Scribd*. https://www.scribd.com/document/506749367/Class-12-Physics-Alternating-Current-and-Electromagnetic-Waves-PPT-3-of-3 (accessed Jun. 10, 2022).

[2] "The Electromagnetic Spectrum - Radio Waves And Radio Spectrum." https://byjus.com/physics/electromagnetic-spectrum-radio-waves/ (accessed Jun. 20, 2022).

[3] "Wave–particle duality," *Wikipedia*. Mar. 24, 2022. Accessed: Jun. 10, 2022. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Wave%E2%80%93particle_duality&oldid=1078913107

[4] "Gamma Rays | Science Mission Directorate." https://science.nasa.gov/ems/12_gammarays (accessed Jun. 10, 2022).

[5] Saad Dahleb University Of Blida Faculty Of Engineering Sciences, Department Of Electronics, Magister's Memory, Speciality: Images And Words, Study And Implementation On DSP, Of Digital Signal Processing Techniques, Used In Nuclear Spectroscopy, By Messai Adnan

[6] D. Reilly, N. Ensslin, H. Smith, S. Kreiner, E.-U. Los Alamos National Laboratory (U.S.), and Nuclear Regulatory Commission, *Passive nondestructive assay of nuclear materials*. Springfield, VA: US Department of Commerce, National Technical Information Service, 1991.

[7] Q. Wang *et al.*, "Multi-channel analyzer based on a novel pulse fitting analysis method," *Nucl. Eng. Technol.*, vol. 54, no. 6, pp. 2023–2030, Jun. 2022, doi: 10.1016/j.net.2021.12.019.

[8] M. D. R. Silva, *Ionizing Radiation Detectors*. IntechOpen, 2015. doi: 10.5772/60914.

[9] J. S. S. Hitch, R. Keyser, and T. Twomey, "Why high-purity germanium (HPGe) radiation detection technology is superior to other detector technologies for isotope identification," tech. rep., Ortec, 2006.

[10] W. Quinn, "University College London Lab 3 Experimental Physics PHAS005," p. 15.

[11] H. Spieler, "Pulse Processing and Analysis, excerpt from IEEE NPSS Short Course Radiation Detection and Measurement," in *IEEE Nuclear Sciences Symposium, San Francisco, California*, 1993, pp. 2–4.

[12] E. W. TAFO, S. U. Hong, P. Yu, W. U. Ming, and Q. Yi, "A low noise charge sensitive preamplifier with switch control feedback resistance," *Nucl. Sci. Tech.*, vol. 19, no. 1, pp. 39–44, 2008.

[13] R. M. Keyser, R. D. Bingham, and T. R. Twomey, "Improved performance in germanium detector gamma-spectrometers based on digital signal processing," *J. Radioanal. Nucl. Chem.*, vol. 276, no. 3, pp. 567–575, Jun. 2008, doi: 10.1007/s10967-008-0601-2.

[14] S. Usman and A. Patil, "Radiation detector deadtime and pile up: A review of the status of science," *Nucl. Eng. Technol.*, vol. 50, no. 7, pp. 1006–1016, Oct. 2018, doi: 10.1016/j.net.2018.06.014.

[15] A. Georgiev and W. Gast, "Digital pulse processing in high resolution, high throughput, gamma-ray spectroscopy," *IEEE Trans. Nucl. Sci.*, vol. 40, no. 4, pp. 770–779, 1993.

[16] P. Babu and P. Stoica, “Sparse spectral-line estimation for nonuniformly sampled multivariate time series: SPICE, LIKES and MSBL,” in *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, 2012, pp. 445–449.

[17] C. Imperiale and A. Imperiale, “On nuclear spectrometry pulses digital shaping and processing,” *Measurement*, vol. 30, no. 1, pp. 49–73, 2001.

[18] V. Radeka, “Trapezoidal filtering of signals from large germanium detectors at high rates,” *Nucl. Instrum. Methods*, vol. 99, no. 3, pp. 525–539, 1972.

[19] V. T. Jordanov and G. F. Knoll, “Digital synthesis of pulse shapes in real time for high resolution radiation spectroscopy,” *Nucl. Instrum. Methods Phys. Res. Sect. Accel. Spectrometers Detect. Assoc. Equip.*, vol. 345, no. 2, pp. 337–345, 1994.

[20] G. L. ZHANG Huaiqiang and G. L. ZHANG Huaiqiang, “Optimal choice of trapezoidal shaping parameters in digital nuclear spectrometer system,” 《核技术》(英文版), vol. 24, no. 6, pp. 60407–060407, Jan. 2014, doi: 10.13538/j.1001-8042/nst.2013.06.011.

[21] Hong Xu *et al.*, “New methods to remove baseline drift in trapezoidal pulse shaping,” *Nucl. Sci. Tech.*, vol. 26, no. 5, p. 5, 2015.

[22] D. H. N. Quy, P. N. Tuan, and N. N. Dien, “Design and Construction of a Digital Multichannel Analyzer for HPGe Detector Using Digital Signal Processing Technique,” *J. Anal. Sci. Methods Instrum.*, vol. 09, no. 02, pp. 22–29, 2019, doi: 10.4236/jasmi.2019.92003.

[23] G. Zeng *et al.*, “Baseline restoration technique based on symmetrical zero-area trapezoidal pulse shaper,” *Nucl. Instrum. Methods Phys. Res. Sect. Accel. Spectrometers Detect. Assoc. Equip.*, vol. 858, pp. 57–61, Jun. 2017, doi: 10.1016/j.nima.2017.03.049.

[24] M. Bogovac, “Digital Pulse Shaping,” p. 57.

[25] “Vivado Design Suite: AXI Reference Guide (UG1037) Ug1037-vivado-axi-reference-guide,” *UserManual.wiki*. https://usermanual.wiki/Document/ug1037vivadoaxireferenceguide.1957650522/html (accessed Jun. 10, 2022).

[26] S. Scholl, “The Xilinx Zynq: A Modern System on Chip for Software Defined Radios,” 2016.

[27] “Welcome to Processing!,” *Processing*. https://processing.org// (accessed Jun. 11, 2022).

ملخص

الهدف من العمل المنجز في هذه المذكرة هو إنجاز مطياف نووي رقمي خاص بالأشعة غاما التي تأتي عادة من التفاعلات النووية. تصميم المطياف يتطلب منا أن نأخذ بعين الاعتبار مختلف الظواهر التي تأثر على دقة القياس كالضجيج الالكتروني، و تراكم النبضات في حالة النشاط الاشعاعي العالي. كل هذا سيكون عن طريق معالجة الاشارة المستقبلة من الحساس الرقمي فتصبح هناك إمكانية تصحيح الأخطاء القياسية للرفع من دقة المطياف. في هذه المذكرة قد تم هذا عن طريق دراسة نظرية خاصة بهذا النظام، ثم انجاز المحاكات على برنامج ماطلاب بالاضافة الى الانجاز التطبيقي لمطياف غاما الرقمي أين تم برمجة لوحة زادبورد التي تحتوي على جزء مصفوفة البوابات المنطقية القابلة للبرمجة و معالج، لمعالجة الإشارة رقميا وتصنيف النتائج المحصل عليها ليتم استقبالها من طرف الحاسوب كي يرسم الطيف الخاص بمصدر الأشعة غاما.

Abstract

The aim of the work in this thesis is to make a digital nuclear spectrometer for gamma rays that are usually emitted during nuclear reactions. The design of the spectrometer requires taking into account the various phenomena that effect the accuracy of the measurement, such as electronic noise and the accumulation of pulses in high radioactivity, by digitally processing the received signal from the sensor. Therefore, it corrects the errors in order to increase the resolution of the spectrometer. This was done in through a theoretical study of the system, the simulation on MATLAB and also the implementation of the digital gamma spectrometer, where the Zedboard board that combine an FPGA and processor was programmed to digitally process the signal and classify the obtained results that will be received later by the computer to display the spectrum of the gamma ray source.

Résumé

L'objectif des travaux de cette thèse est de réaliser un spectromètre nucléaire numérique pour les rayons gamma habituellement émis lors des réactions nucléaires. La conception du spectromètre nécessite de prendre en compte les différents phénomènes qui affectent la précision de la mesure, tels que le bruit électronique et l'accumulation d'impulsions à haute radioactivité, En traitant numériquement le signal reçu du capteur, donc il corrige les erreurs afin d'augmenter la résolution du spectromètre. Cela a été fait dans ce travail à travers une étude théorique du système, la simulation sur MATLAB et aussi l'implémentation du spectromètre gamma numérique, où la carte Zedboard qui combine une partie FPGA et des processeurs a été programmée pour traiter numériquement le signal et classer les résultats obtenus qui seront reçus par l'ordinateur pour afficher le spectre de la source de rayons gamma.

Key Words

Digital spectroscopy systems, Digital Trapezoidal shaper, Multichannel analyzer, Maxima detection, Gamma ray spectrum.