



Politechnika Łódzka



POLITECHNIKA ŁÓDZKA

Wydział Elektrotechniki, Elektroniki, Informatyki i Automatyki Politechniki Łódzkiej

### Praca Dyplomowa Magisterska

**Cyfrowe przetwarzanie impulsów z detektorów  
promieniowania w czasie rzeczywistym z  
wykorzystaniem układów programowalnych FPGA**

**Real Time Digital Pulse Processing from Radiation  
Detectors Using Field Programmable Gate Arrays**

**inż. Wojciech Mateusz Walewski**

Nr albumu: 239587

*Promotor:*

dr hab. inż. Dariusz Makowski, prof. uczelnii

14 sierpnia 2022

# Contents

<b>1</b>	<b>Introduction</b>	<b>6</b>
1.1	Motivation . . . . .	6
1.2	Fission energy . . . . .	7
1.3	Fusion energy . . . . .	8
1.4	ITER Tokamak Project . . . . .	11
1.4.1	Tokamak . . . . .	11
1.4.2	ITER Goals . . . . .	11
1.5	Real time diagnostics . . . . .	12
1.5.1	Field Programmable Gate Arrays . . . . .	13
1.6	Problem statement . . . . .	14
<b>2</b>	<b>Hard X-Ray spectroscopy</b>	<b>15</b>
2.1	Runaway electrons . . . . .	15
2.2	PhotoMultiplier Tubes . . . . .	16
2.3	Preamplifiers . . . . .	17
2.4	Pulse processing chains . . . . .	17
2.4.1	Analogue processing chains . . . . .	17
2.4.2	Digital processing chains . . . . .	18
2.5	ITER HXRM specification . . . . .	19
<b>3</b>	<b>Research setup</b>	<b>21</b>
3.1	PMT . . . . .	21
3.2	Preamplifier . . . . .	21
3.3	Digitizer board . . . . .	22

3.3.1	Open FPGA design . . . . .	23
3.3.2	Parallel sampling . . . . .	23
3.4	Host computer . . . . .	24
<b>4</b>	<b>Pulse detection</b>	<b>25</b>
4.1	Level trigger . . . . .	25
4.2	Boxcar filter . . . . .	27
4.3	Trapezoidal filter . . . . .	29
4.4	Triangular filter . . . . .	32
4.5	Other pulse detection methods . . . . .	33
4.6	Simulation performance . . . . .	33
<b>5</b>	<b>Pulse Height Analysis</b>	<b>36</b>
5.1	Pulse shaping algorithms . . . . .	36
5.2	Integration . . . . .	36
5.3	Simulated performance . . . . .	37
<b>6</b>	<b>HXRIM firmware</b>	<b>38</b>
6.1	System overview . . . . .	38
6.2	System control . . . . .	39
6.3	Pulse detection . . . . .	40
6.4	Pulse height analysis . . . . .	43
6.4.1	Experimental results . . . . .	44
6.5	Spectrum storage and transfer . . . . .	44
<b>7</b>	<b>Data transfer</b>	<b>47</b>

7.1	PCIe interface . . . . .	47
7.2	Direct memory access . . . . .	48
7.3	Acquisition software . . . . .	49
7.3.1	Multi-threaded data transfer . . . . .	51
7.3.2	Processing routine optimization . . . . .	53
7.3.3	Write speed optimization . . . . .	54
<b>8</b>	<b>Conclusions</b>	<b>57</b>
8.1	Further problems and research . . . . .	58

## **Streszczenie**

  Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

  Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

  Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

## **Abstract**

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

# 1 Introduction

## 1.1 Motivation

Concerns regarding the sustainability of using fossil fuels for energy generation have been raised in as early as the 1970s [1]. One of the most well-known examples from that time was the 1972 report titled "Limits to Growth" by Meadows et. al. [2]. In it a group of MIT scientists attempted to answer the question of how long will the Earth's natural resources last for, considering the seemingly neverending growth of human civilisation. As a result of a conducted computer simulation, a rough estimate of around 100 years was given as the deadline. After that time the population would start to collapse due to the lack of resources.

This estimate did not go without controversies back when it was first published. The methodology was thoroughly picked apart, leading many to dismiss the study findings [1]. Naturally, nowadays, we are much better suited to verify the claims made by the now 50 year old book. The impeding resource depletion has certainly been made a less valid claim as technological progress made it possible to locate and tap into previously inaccessible resource fields [3]. Taking into account other issues, however, the original timeline of 100 years might have actually shifted closer.

When it comes to fossil fuel usage, in the last twenty years, the primary concerns have changed from resource depletion to global warming and irreversible environmental damage [1]. In 2018 the Intergovernmental Panel on Climate Change (IPCC) published a report indicating the need to stop the average global temperature at  $1.5^{\circ}\text{C}$  above the levels measurable in the pre-industrial era. The panel projected that failure to do so, would lead to irreversible climate changes and in turn serious damage to human settlements around the world. [4]

Fossil fuels account for as much as 70% of greenhouse gas emissions. Electricity generation alone causes 25-35% [5]. Such a high share means that reducing this output is crucial in meeting the goals outlined by the IPCC. At the beginning of the twentieth century, renewable energies, i.e. wind, solar, biomass and geothermal were thought to be the perfect solution to the issue at hand [6].

In modern times, we have now become aware of multiple issues that make renewable energy generation a problem at a larger scale. Most importantly, their efficacy varies depending on the geographical location and climate. Even when placed in optimal con-

ditions, they do not offer perfect stability. Additionally, the land usage is greater than with the more traditional forms of energy production [7].

## 1.2 Fission energy

In recent years, the drawbacks of renewable energies have produced an alternative approach in both research and policymaking. The use of nuclear energy for supplementing the shortcomings of renewables has been suggested as a potential path forward. This concept is sometimes referred to as hybrid nuclear-renewable system. [8].

There are two ways that nuclear energy can be created and harnessed. In the more well-established technology, fission, heavy atoms (usually Uranium) are bombarded with neutrons and split into two or more lighter nuclei and additional neutrons as shown in Figure 1. The reaction is self-sustaining and releases energy in the form of heat that is then used to boil water. The produced steam then causes turbines to spin and generate electricity.

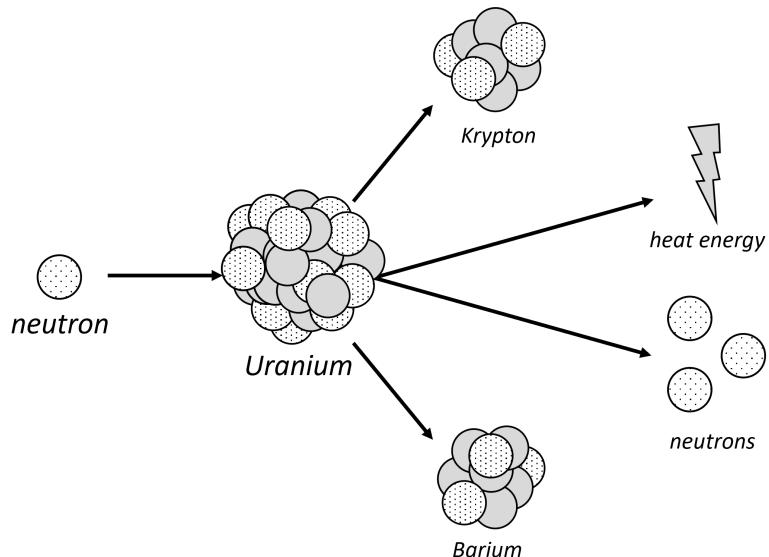


Figure 1: Uranium fission reaction

Fission is far from a new concept, as first fission reactors have been built in as early as 1942 [9]. Although the technology itself is quite old and has been greatly improved over time, there is reasonable reluctance to build and use fission power plants. The issue that gets raised most often is the storage of radioactive waste. There are, however, multiple less well-known problems with fission [10].

The tragedies of Chornobyl and Fukushima reactors have caused many people to be wary of fission. Now, even if societal support would be disregarded in policymaking, the acquisition, storage and disposal of radioactive materials required for and produced during fission proves to be an administrative challenge. This is particularly problematic if reactor construction and maintenance is to be handled by private entities [11]. The complexity of the problem with energy production suggests that as we arrive to more concrete solutions we should not stop exploring other potential alternatives.

### 1.3 Fusion energy

Just like it is possible to split atoms, it is also possible to combine them together in a process known as fusion. Fusing atoms lighter than Iron is a reaction that can produce surplus energy as highlighted by Figure 2. Two atoms of low binding energy can fuse into an atom of higher binding energy, releasing heat based on the mass-energy equivalence formula. The output energy can be used to generate electricity in the exact same manner as with fission. The problem with fusion reactors is that the conditions necessary for fusion to happen are extremely harder to achieve and sustain [12].

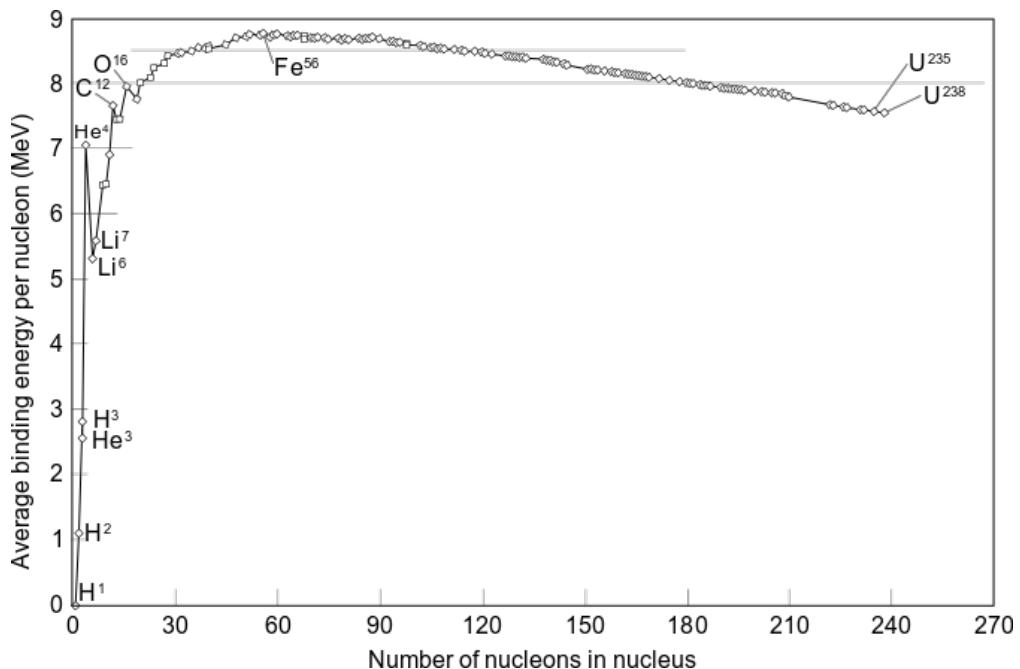


Figure 2: Average binding energy per nucleon as a function of the number of nucleons in the atom

Fusion is the primary reaction that causes stars to emit light and heat. The reaction that is most often artificially attempted on Earth differs from that occurring naturally in the

Sun. There, a p-p reaction occurs: 4 protons into  ${}^4\text{He}$ . The p-p reaction is not the most efficient but the Sun's gravitational field makes it feasible. On Earth, fusion experiments primarily rely on using hydrogen isotopes, most commonly deuterium (D) and tritium (T) as shown in Figure 3. This is a considerably simpler reaction that offers the most promise when it comes to fusion devices.

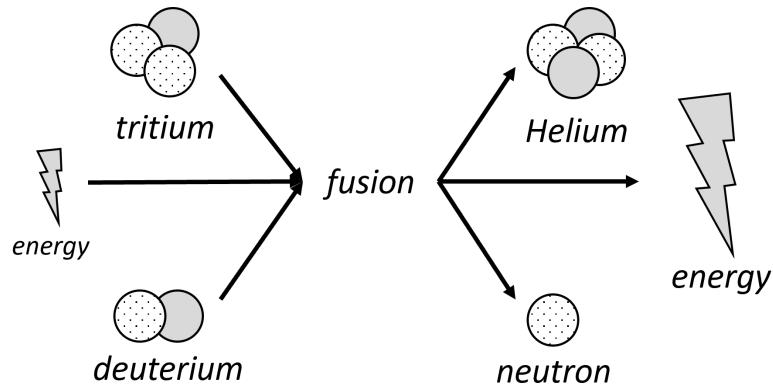


Figure 3: D-T fusion reaction

Despite being an easier approach, it still requires the reactor to sustain a 200 million °C plasma. This means that a large amount of energy must be used to first heat the plasma up and then confine it to prevent it from completely destroying the reactor. The efficiency of D-T reactions might, however, be worth the trouble. Theoretically, just 30 mg of deuterium would generate as much energy as 250 l of gasoline [13].

To achieve this output a few hurdles must first be overcome. Tritium, the other input material of the D-T reaction is extremely rare in nature. Its artificial production is currently done only by a select number of facilities. Combined with its relatively short half-life of around 12 years, concerns regarding tritium shortage have recently been raised. Future fusion reactors like, should be able to "breed" their own tritium, however the transition period may still prove to be troublesome [14].

In the end, despite being a similarly old technology as fission [15], a fusion reactor with a net positive energy balance has not yet been constructed. Containing plasma heated to such extreme temperatures cannot be achieved with any solid material and must be done with the use of inertial or magnetic forces. The most common reactors that rely on magnetic confinement idea are: tokamaks (Figure 4) and stellarators (Figure 5). [16]

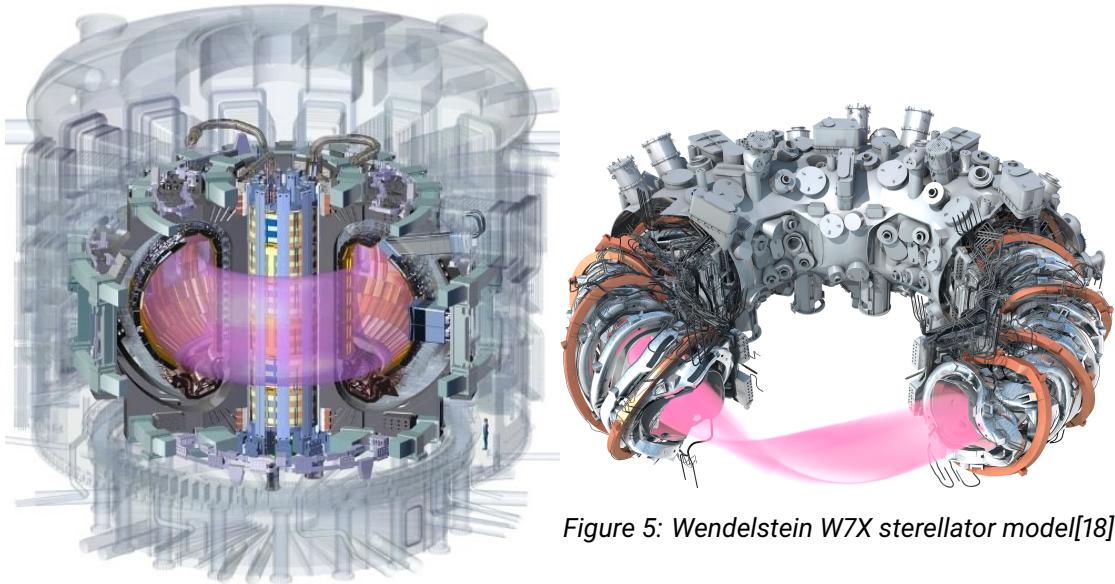


Figure 4: ITER tokamak model[17]

Figure 5: Wendelstein W7X stellarator model[18]

Stellarators and tokamaks share some common features, and have been discovered within a few years of each other. In the 20th century the stellarator design proved to be more complex to construct than the tokamak design. For that reason significantly more research was done with tokamaks. Although interest in stellarators has resurfaced recently with the Wendelstein W7X project, the International Thermonuclear Experimental Reactor (ITER) is going to be a tokamak. [13]

## 1.4 ITER Tokamak Project

### 1.4.1 Tokamak

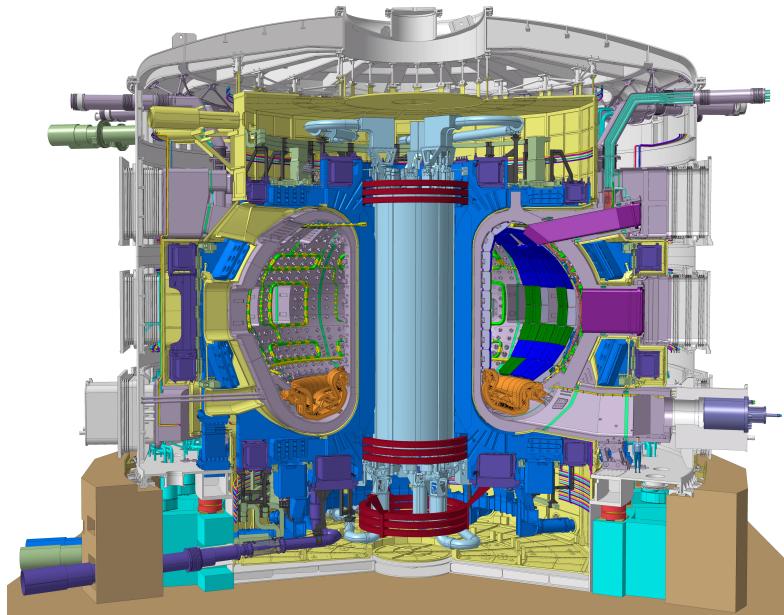


Figure 6: ITER tokamak cross section[17]

Tokamaks are reactors composed of a toroidal vacuum chamber surrounded by massive electromagnetic coils as shown in Figure 6. The magnetic fields are used to shape plasma and keep it away from the device internal walls. Operation starts with the removal of air and impurities from the vessel. The gaseous fuel is introduced and a strong current is induced ionizing the gases, thus forming plasma. Additional heat is introduced with microwaves and fuel injections. After a specific temperature is reached, particles within the plasma can collide with such force that they begin to break through atomic repulsion and fuse producing a large amount of energy. [17]

### 1.4.2 ITER Goals

At 24 m height and 30 m width, ITER will be the biggest tokamak ever constructed. That's twice as big as the largest experimental reactor currently in service, the Joint European Torus (JET). JET has been working since 1983 with a goal of achieving a net energy gain. Despite running successful plasma experiments and recently starting deuterium-tritium experiments, the highest Q (the ratio of produced power to the power required to sustain

the plasma) obtained by JET was 0.33. A record only recently topped by a different type of a fusion device at the US Department of Energy's National Ignition Facility. Using laser technology it was possible to obtain a Q of 0.7 for 4 billionths of a second. [19]

Although not supposed to work as a power plant, throughout its operation ITER intends to break the record by a lot. The new reactor is designed to reach an energy gain of 10 for the duration of a few minutes. This is still a theoretical plan, as the facility is still a long time from being finished. As of 2022, the vacuum vessel must still be welded together[20]. First plasma is scheduled for the end of 2025[20], although multiple delays have already happened in the past [21], so future ones would, at this point, not be too surprising. Figure 7 shows an aerial picture of ITER from 14th of April 2020.



Figure 7: Aerial view of the ITER facility as of 2022[17]

## 1.5 Real time diagnostics

With a project of such large magnitude as ITER it is near impossible to predict everything that might go wrong. There are naturally safety concerns, questions regarding the ambitious goals set by the management and the troubles of international cooperation[21]. It is thus crucial to maintain maximum operational safety and log all experimental data. This should ensure, that even in the case of partial failure of the project, important practical knowledge can be recorded for future fusion experiments.

When operational, ITER will rely on around 50 completely different measurement systems to control, evaluate and investigate its plasma [22]. This translates into dozens of gigabytes of data being generated, processed and archived every second as the experi-

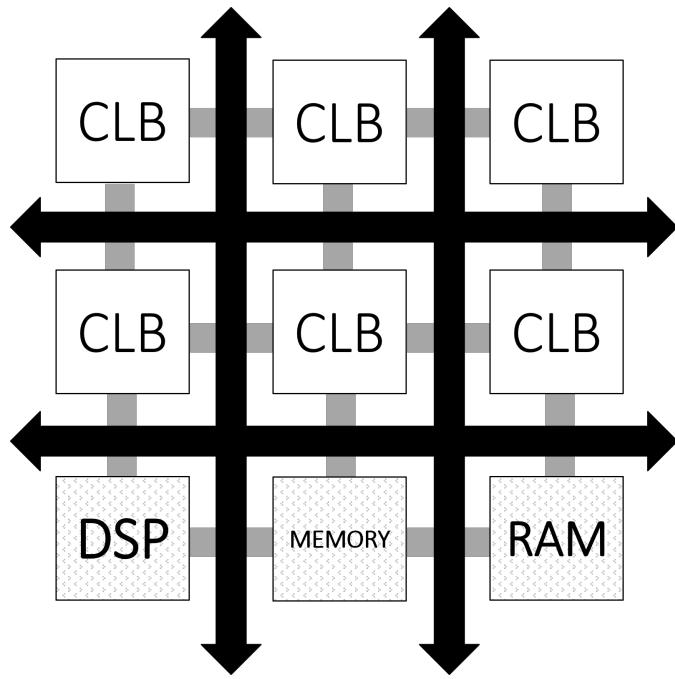
ment is running [23]. With some of the experiments lasting fractions of a second, a lot of the critical data acquisition and control must occur in real time and without waiting for human input [24]. This poses an important challenge when it comes to the choice of computing apparatus. A perfect device would offer infinite configurability preferably with remote access as well as a very high data throughput.

Traditional computers or more precisely Central Processing Units (CPUs) are remotely reprogrammable but may offer insufficient speeds in some of the real time applications. On the other end of this scale lie Application Specific Integrated Circuits (ASICs). These devices are an arrangement of digital logic gates realizing one specific goal, like digital signal filtering or processing network packets. ASICs offer unmatched speeds but cannot really be reconfigured after deployment, making them a risky choice in highly experimental applications such as tokamaks.

### 1.5.1 Field Programmable Gate Arrays

Devices that offer a compromise between speed and reconfigurability exist nowadays. Most commonly Field Programmable Gate Arrays (FPGAs) are used. FPGAs are formed out of matrices of Configurable Logic Blocks (CLBs). CLBs are small circuits that produce a single bit of logic output out of 4 input bits, using a reprogrammable function. These blocks are then wired together using programmable interconnects. A finite combination of CLBs can realise any digital function. Such design allows for the implementation of very efficient algorithms directly with the use of logical gates, without the need for an entire processor. [25]

The interconnects and CLB internal structure introduce additional wiring that would be unnecessary in the case of ASICs. These stray capacitances and inductances limit the maximum clock speed of FPGAs. This is not too much of an issue provided that the function can be parallelised and pipelined. In addition, to offset this limitation, most FPGAs come packed with more complex sub-circuitry that can be intermixed with the CLBs as indicated in Figure 8. These components can vary greatly from efficient arithmetical blocks to small CPU-based microcontrollers.



*Figure 8: The internal structure of an FPGA*

## 1.6 Problem statement

The nature of FPGAs makes them instrumental in data acquisition and control systems of modern fusion experiments like the ITER tokamak. This work evaluates the usage of Field Programmable Gate Arrays for the detection and analysis of pulses produced by PhotoMultiplier Tubes (PMTs) being part of a High X-Ray Monitor (HXRM) designed to monitor Runaway Electrons (REs) in tokamaks.

A functional system for the Digital Pulse Processing (DPP) at the rate of 1 GS/s is proposed and implemented in an FPGA. A custom software package for control and data acquisition is shown. The optimization of both hardware and software is described. Different algorithms for pulse detection and discrimination are analysed, simulated and implemented in hardware. Experimental results produced by the system are described. Finally design considerations for future systems are given.

## 2 Hard X-Ray spectroscopy

One of the diagnostic systems present in most of the currently existing tokamaks is the Hard X-Ray Monitor (HXRM). It will also be used in ITER. The device is tasked with measuring the spectrum of X-Ray radiation inside the fusion vessel. The presence of high energy X-Ray radiation can point to problems with the plasma stability and suggest the need for mitigation techniques. [26]

### 2.1 Runaway electrons

The plasma in a tokamak is super-heated so that particles reach a velocity that enables them to break through atomic repulsion. In such conditions some particles can also obtain sufficient speed to escape the vessel's magnetic confinement. Typically collisions with other particles are so frequent that in stable plasma these effects are rare. Directly after plasma is disrupted or terminated, the probability of collisions is lessened and a tokamak might start behaving like a particle accelerator, accelerating the electrons to nearly the speed of light. Electrons that act in this manner are called Runaway Electrons (REs). [27]

This phenomenon can occur in the form of a high-energy concentrated beam capable of melting the front-facing walls of a reactor. The effects of such damage being purposefully introduced in the JET tokamak are shown in Figure 9. To prevent and mitigate the destructive effects of REs their generation has to be avoided if possible. Otherwise they must be detected and dealt with. Typically the plasma is cooled, or the electromagnets are accentuated to reduce the disruptions. More extreme methods might involve the injection of noble gases in a process called Massive Gas Injection (MGI) [28].

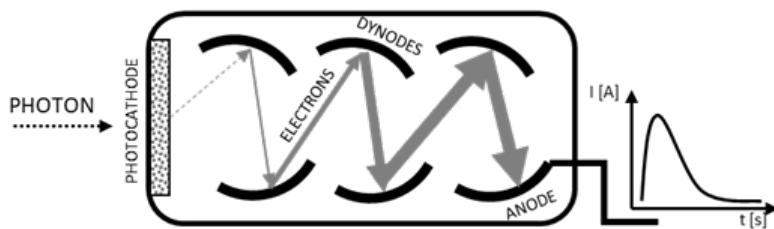


*Figure 9: Effects of RE in the JET vessel[27]*

When REs interact with the Plasma Facing Components (PFCs) they lose their energy and emit X-Ray radiation in the Bremsstrahlung process. The energy of this radiation varies greatly, ranging from tens of keV (Soft X-Ray) to multiple MeV (Hard X-Ray). [29]

## 2.2 PhotoMultiplier Tubes

The photons generated in Bremsstrahlung by Runaway Electrons can be detected with the use of a device known as a PhotoMultiplier Tube (PMT). A PMT is built with the use of a photocathode and an electron multiplier. When a photon hits the photocathode an electron is emitted due to the photoelectric effect. The electric fields inside the PMT accelerate the electron towards a series of dynodes. Each collision with a dynode releases additional electrons and forms a stronger beam that eventually reaches the anode, where it becomes measurable as a current pulse. Figure 10 shows the internal structure of a PMT together with the output signal, a sharp exponentially decaying current pulse. [30, 31]



*Figure 10: A PhotoMultiplier Tube*

## 2.3 Preamplifiers

The internal gain of a PMT, obtained from the electron multiplication is sufficient for many applications. The produced current impulses can be detected with precise circuitry, however there are multiple reasons for which a preamplifier is often used in tangent with a PMT. With a typical load of  $50\Omega$  the output signal of a single photon is a very sharp voltage peak of around 10 mV. These short pulses can be detected with precise instruments and used for counting and timing purposes, but may be problematic when it comes to discrimination and more advanced processing like Pulse Height Analysis (PHA). [32]

In those scenarios that do not require the sharpest response, the slight increase in Signal to Noise Ratio (SNR) is worth the features introduced by an amplifier. These can include impedance matching, filtering and pulse-shaping. In fusion reactors it also helps move most of the diagnostic infrastructure away from the difficult environment created by the fusion plasma. Long wiring is susceptible to ElectroMagnetic Interference (EMI), so the millivolt signal produced by a PMT would get drowned out in the plasma-induced interference without amplification.

## 2.4 Pulse processing chains

To obtain a radiation spectrum from the voltage pulses, their height must be measured and placed into an appropriate bin consisting of a range of voltage levels. The pulses generated by a PMT last just a few nanoseconds, making the task at hand complicated. With a preamplifier this duration is increased by a value that depends on the preamplifier components. This results in pulses that last a few hundreds nanoseconds. Before the advent of ultra-high speed Analogue to Digital Converters (ADCs), such short events could not viably be processed with digital electronics and had to rely on analogue components.

### 2.4.1 Analogue processing chains

In analogue radiation spectroscopy pulses are typically first transformed to a Gaussian shape, with a series of low- and high-pass filters. These signals must then pass through complicated pile-up rejection circuitry. After that the pulses are fed to a Multi Channel Analyzer (MCA). This is the device that performs the binning action. Initially an MCA

would consist of an array of analogue comparators, and over time it would rely on more and more digital components. The schematics of a typical analogue system are shown in Figure 11

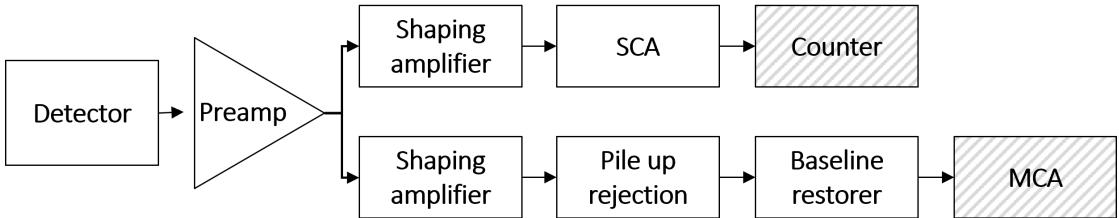


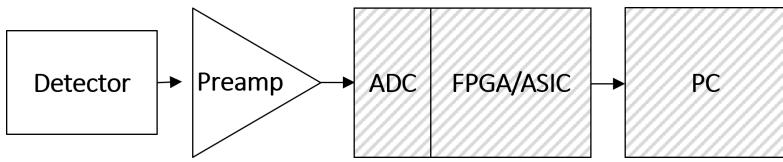
Figure 11: Analogue Pulse Processing chain

As mentioned earlier, for a long time analogue processing was the only way to reliably handle events shorter than 100 ns. It was, however, quickly recognized that a more digital approach offered much lesser susceptibility to outside noise. Digital components could also potentially be tuned without having to physically modify the circuit. These two features are particularly important in the complicated environment of a fusion reactor. [33]

#### 2.4.2 Digital processing chains

As soon as ADCs and digital processing circuits capable of reaching the resolution required for precise nuclear spectroscopy appeared on the market, they were adopted into new experimental designs of MCAs [34]. As the technology improved, ADCs were moved closer to the radiation detector itself. A single reprogrammable silicon chip, together with a fast ADC, could perform the job of multiple analogue components in a spectroscopy system. On top of that, its operation is less susceptible to Electro Magnetic Interference (EMI) and temperature-induced parameter variance. [35]

The earlier in a processing chain that the ADC is placed the lesser the influence of the imperfect analogue components is. An ADC being inserted right after the preamplifier is commonplace in modern systems as shown in Figure 12. Such approach, produces an important issue. To obtain a sufficient horizontal resolution, comparable to analogue systems, the ADC must sample the signal with a frequency of at least a few hundred MHz. This means that, at a typical vertical resolution of anywhere between 8 to 16 bits, a modern high-speed ADC can generate anywhere between 100 megabytes up to even a few gigabytes of data every second. [35]



*Figure 12: Digital Pulse Processing chain*

Processing gigabytes of data in real time poses one of the primary challenges in designing Digital Pulse Processing systems. Despite that, fast digitizers are used for Hard X-Ray Spectroscopy in existing tokamaks, like KSTAR or JET [29, 36]. ITER will require similar or better systems during its operation, so the problems of handling large data throughput should be considered solved before the diagnostic systems are installed on site. Once first plasma is obtained, the possibility of hardware modifications will be greatly limited.

## 2.5 ITER HXRM specification

The Hard X Ray Monitor planned to be installed at ITER will have to meet very specific requirements. The device has to operate in two modes. The first mode is the pulse counting mode. In it the pulses produced by the PMT have to be detected, categorized and binned. This actions has to be performed continuously over a configurable time period between 1 ms and 10 ms. Each sampling window must then produce an energy spectrum of the collected pulses. [37]

Although storing the entire spectrum is useful for research purposes, two figures must be transferred as quickly as possible to the Plasma Control System (PCS), so that critical decisions regarding the tokamak's safety can be made. The PCS requires the maximum runaway electron energy measured and the total count to operate. This data has to be made available with a latency lesser than 5 ms after each sampling window. [37]

When the number of runaway electrons becomes overwhelming for the PMT detector, the signal produced by the device will appear closer to a constant current, due to the pulse overlap. When that scenario occurs the HXRM must switch into its second mode of operation, the current mode. In this mode the PCS must be sent the electric current magnitude measured over the same time window. It is expected that a switch to current mode will be made when the number of pulses exceeds  $10^6$  counts per second.

On top of the spectral information sent to PCS, the device should also transfer unpro-

cessed (raw) samples with the detected pulses to the Data Archiving Network (DAN) for offline analysis and diagnostics. Table 1 lists the most important parameters that ITER's HXRM must meet. The table includes parameters related to operation in both counting and current modes.

*Table 1: Selected ITER's requirements for the HXRM [37]*

Parameter	Value
<i>RE Energy Range</i>	0.1 MeV to 100 MeV
<i>Required energy resolution</i>	20%
<i>RE current to be detected</i>	10kA to 15 MA
<i>ADC sampling period</i>	2ns
<i>Number of ADC channels</i>	2
<i>Max. detector temperature</i>	240°C ± 10°C
<i>Max. magnetic field</i>	5 T

To meet the requirements regarding temperature and magnetic field, as well as the space constraints, some important considerations were taken. In ITER the X-Ray radiation will first be converted to a light pulse with a scintillator. Only the scintillator will be exposed directly on the front wall. The PMT detector will be connected to the scintillator via a 12 m long optical fiber. The PMT will then interface to the processing system through a 100 m long coaxial cable. [38]

The attenuation of fiber optic is usually specified in dB/km so 12 m is a length insufficient to generate meaningful light loss. The critical issue lies in the connectors. Mis-coupling of fiber optic connectors can generate a massive signal loss. In a tokamak's environment it will be nearly impossible to keep the connectors perfectly aligned due to vibrations, temperature variation and magnetic fields. It is thus likely that just a few promilles of the photons generated by the scintillator will actually reach the PMT. [38]

### 3 Research setup

An FPGA powered data acquisition board was used as the platform for the hardware implementation of digital pulse processing of PMT signals. The following subsections describe the system components in detail and Figure 13 shows an overview of the test bench. A 25 kBq  $^{137}Cs$  radioactive sample was placed in a plastic V-vial and used to test the setup.

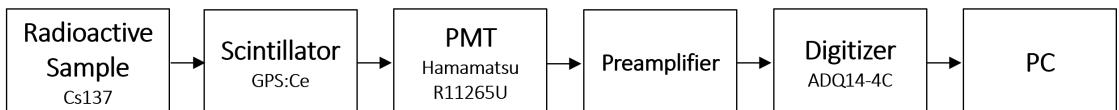


Figure 13: HXRM test bench overview

#### 3.1 PMT

Hamamatsu R11265U served as the PhotoMultiplier Tube. A 15 mm GPS:Ce scintillator wrapped in aluminium foil was coupled directly with the PMT to generate light pulses from the radiation. The PMT with the scintillator and radioactive material were placed in a dark container and held together with a custom-designed 3D printed casing.

#### 3.2 Preamplifier

The design of a preamplifier capable of transforming the PMT signal to satisfy ITER's requirements is a non-trivial task. No commercially available amplifiers were found that would satisfy all of the requirements. Specifically for this project, a custom low-noise preamplifier has been developed by Nowakowski et al. and described in another paper. [26]

The PMT-Preamplifier combination used in this work generates pulses that fit a single exponential pulse curve with a decay constant of 130 ns. Bi- and tri-exponential models grant a higher accuracy, but are were found too problematic for real-time processing.

### 3.3 Digitizer board

To acquire and digitize signals, produced by the PMT and preamplifier combination, Teledyne SP Devices ADQ14-4C was used. The board was connected through PCIe 2.0 to the host PC running custom acquisition software.

Table 2: Chosen parameters of ADQ14-4C

Parameter	Value
<i>Channel count</i>	4
<i>Sampling rate</i>	1 GS/S
<i>Vertical resolution</i>	14 bits
<i>Interface</i>	PCIe 2.0 x8
<i>Max. data transfer rate</i>	3.2 GB/s
<i>Internal DRAM memory</i>	2 GB
<i>Input range peak to peak</i>	0.2 V - 10 V
<i>Number of GPIO ports</i>	12

ADQ14-4C can sample signals from up to 4 channels, each at a maximum frequency of 1 GHz. It acquires samples with a 14-bit ADC. The device applies factory calibrated digital gain and shift to the raw measurements, so in order to maintain a higher precision the samples are extended to 16 bits representing two full bytes. The additional two bits represent the fractional part sometimes produced by the fractional gain component.

The ADQ14 can operate in both triggered streaming and continuous mode. In continuous mode samples are constantly gathered and periodically transferred to the host PC. In triggered streaming a window of samples is collected only after a trigger event is detected. This is a basic feature that allows for some data reduction, as only events of interest have to be transferred to the host PC. Multiple triggering mechanisms, like level, periodic and external are available.

In all modes of operation the device relies on an internal 2 GB DRAM to act as a FIFO queue for the generated records. The device relies on Direct Memory Access (DMA) to transfer data to the host PC. This is a special mode of operation for peripheral devices in which a chunk of the computer's memory is made available to them without the need of CPU brokerage.

Unfortunately, the maximum size of DMA buffers is limited. With a sampling speed of a 1 GHz and 2 byte samples, up to two gigabytes of data can be generated each second

for each channel that is active. Even with reliance on DMA, maintaining a transfer speed this high is problematic. To solve this issue, the ADQ14's internal DRAM acts as a buffer. Records are first stored in the internal DRAM and periodically transferred to the host PC's RAM whenever DMA buffers become available. At maximum speed a single channel can, however, still fill the entirety of this internal buffer in just a second if its contents are not flushed in time.

### 3.3.1 Open FPGA design

A crucial feature of ADQ14 is the fact that its core processing functionality is realised with the use of an on-board FPGA, more specifically a Xilinx Kintex 7 K325T. The design of the FPGA is partially open. Users can implement custom filtering or data analysis on samples in real time. This fact is used to implement the real-time pulse processing described in this work.

The firmware is unfortunately not entirely open-source. Third-party IP cores cannot be distributed to end customers and thus user algorithms are limited to two sections called User Logics. User Logic 1 is a core placed after the ADC samples are subjected to factory gain, but before the signals are passed on to trigger control. This enables the implementation of custom triggering logic.

## [OBRAZEK]

User Logic 2 is intended to house more complicated logic. This module has access to the GPIO ports and some metadata outputs, that can be used to describe the transferred data. User Logic 2 is located right before the encrypted packet generator that is responsible for queueing the incoming data in ADQ14's DRAM for transfer to the host PC. The packet generator can be partially controlled from within User Logic 2. Arbitrary data can be injected in place of the samples for each channel and the size of transferred windows can be modified.

### 3.3.2 Parallel sampling

The FPGA is clocked only at 250 MHz which is exactly a quarter of the ADC maximum operating frequency. This means that each channel of the digitizer produces 4 samples on each clock cycle of the FPGA. This is a necessary design choice as FPGAs fare

better at lower clock speeds due to the need of less complex routing when it comes to connecting the various peripherals and CLBs.

Such design does, however, complicate the implementation of Digital Signal Processing. It is especially cumbersome for functions that depend on delayed samples. Accumulators have the need of summing up 4 samples on each clock cycle instead of one. Complicated operations like multiplication and division require duplicated logic. Most functionality has to be properly pipelined to avoid timing issues in the FPGA design.

### 3.4 Host computer

The host computer was running Red Hat Enterprise Linux 7. Table 3 lists the computer hardware configuration.

*Table 3: Hardware configuration of the host PC*

Parameter	Value
CPU	Intel Xeon C5549
Storage	Samsung 970 EVO Plus 500 GB
Operating System	Red Hat Enterprise Linux 7.4
GPU	NVIDIA GP107
RAM	24 GB DDR3 1333 MHz

## 4 Pulse detection

In order to process pulses they must be first detected. The window of interest containing a pulse to be analysed must be properly distinguished from the surrounding noise. Although detection is a must, it also brings the added benefit of data rate reduction. If pulses can be accurately marked within a signal it is possible to transfer only those samples that compose the events to the host PC. The remaining parts of signal that contain only the thermal noise can be omitted.

The ADQ14 is capable of sampling signals with a frequency of up to 1 GHz. With two bytes per sample, a single channel generates 2 GBs of raw data per second. For the PCIe 2.0 interface, that is used in this work, the manufacturer's datasheet specifies a theoretical maximum throughput of 3.2 GB/s. Even if this value could be obtained it would not allow for two channels to be active at once. Using pulse detection for data reduction is thus unavoidable. It is crucial to use a robust algorithm for this process to ensure that no pulses go unnoticed and near to none false positives are transmitted.

### 4.1 Level trigger

The most basic approach to detection is a level trigger, a technology available in virtually any spectrometer. As shown in Figure 14, a trigger occurs when the input signal crosses a predefined voltage level, either on a rising or a falling edge. The point at which this happens marks the beginning of a record window. In the simplest case the end of a window is placed a fixed duration from the start. Samples within that window form a region of interest and are transferred further down the processing pipeline.

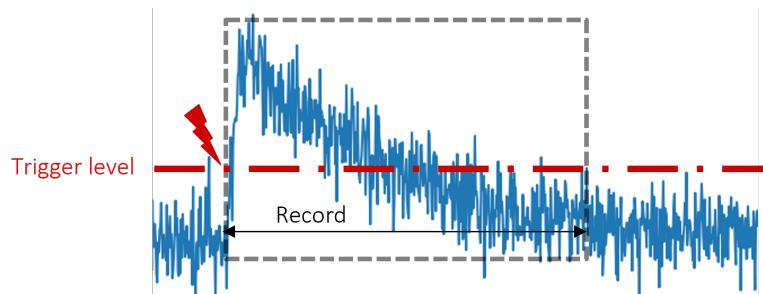
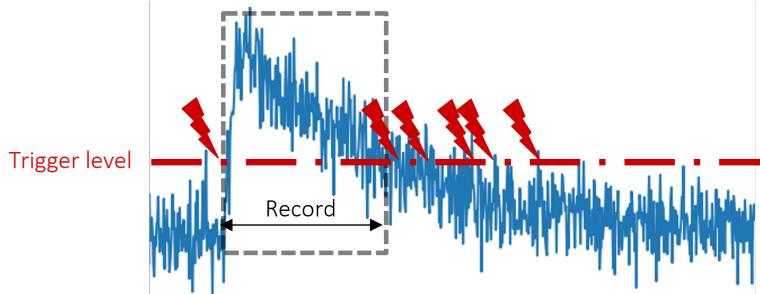


Figure 14: Level trigger

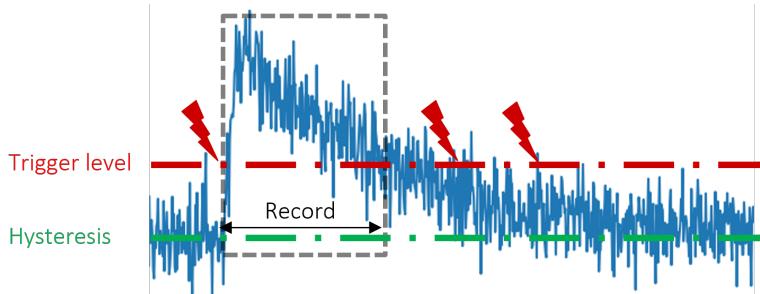
Once a window finishes, no more events are detected until the signal returns to a value below the trigger level (for rising edge triggering). This reset value might be set to be

equal to the trigger level, however such approach might lead to a scenario shown in Figure 15. In a noisy environment the signal might falsely trigger immediately after a window ends due to a random high spike on the slower falling edge of an exponential pulse.



*Figure 15: Level trigger with no hysteresis*

To prevent such false triggers typically some form of a hysteresis is used. The reset level is shifted downwards, so that the input signal must cross a second threshold before the another trigger can be raised. This threshold should be set to such value, that random noise is guaranteed to never cause false triggers, but not any lower. Figure 16 shows a hysteresis mitigating some false triggers. Some of the false triggers from Figure 15 are removed, however due to the high noise some still persist.



*Figure 16: Level trigger with a hysteresis*

With the slow falling edge of a pulse, care must be taken not to overestimate the hysteresis. By setting the reset level too far away from the trigger level pile-ups can be missed as pointed in Figure 17. With a properly set hysteresis the level trigger offers performance that is sufficient in most applications. In a tokamak's environment, however, the hysteresis alone could potentially prove insufficient due to EMI and temperature fluctuations.

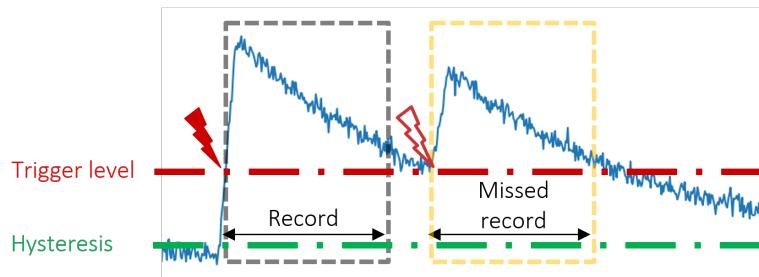


Figure 17: Level trigger hysteresis causing missed events

## 4.2 Boxcar filter

The two problems inherent to simple level triggering are its susceptibility to noise and pulse overlap. Just as with analogue approaches these two problems can be minimized with low-pass and high-pass filtering. By utilizing a low-pass filter the input signal becomes smoother, reducing the possibility of false triggers caused by random noise spikes. A high-pass filter converts the falling edge of a pulse to a sharper curve, making pile-ups easier to detect.

In the digital domain the simplest filters that perform these operations are the Moving Average (MA) filter and the derivative filter. MA works to reduce the spikes and increase SNR. The derivative filter can strip the DC component from a signal, meaning that the sharp rising edges will become even easier to detect in comparison to the decaying tails.

By combining the concepts of the MA and derivative filters into a single module a boxcar filter is obtained. Figure 18 shows an example boxcar transfer function, together with its effect on an input signal of two exponential pulses. The boxcar filter can be considered to be a subtraction of two samples averaged with a window of length  $W$  delayed from each other by  $W$ . On Figure 18  $W = 25$ .

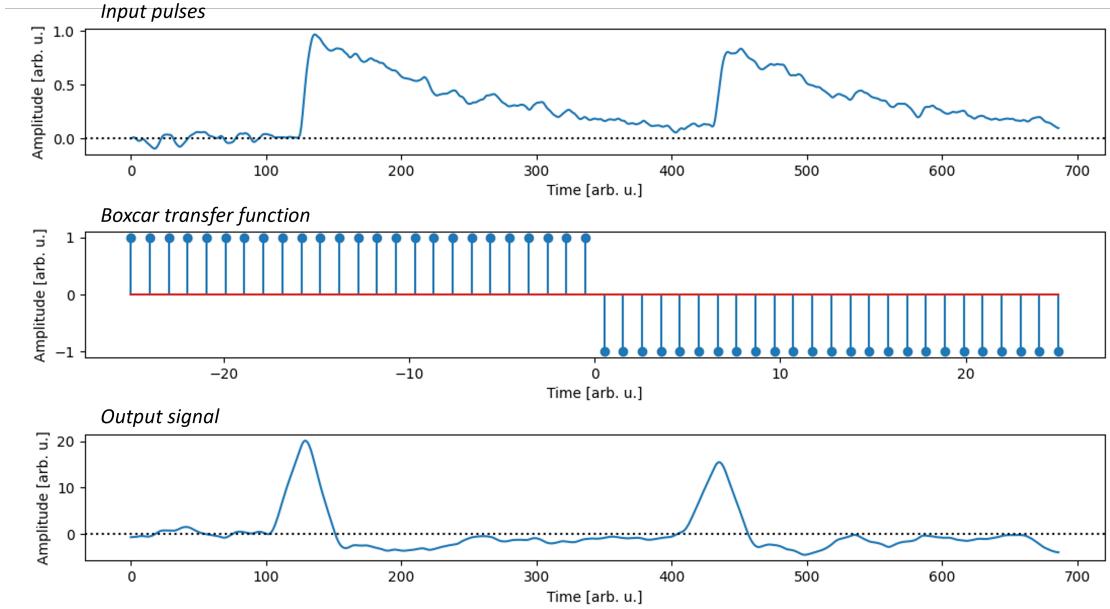


Figure 18: Effect of boxcar filtering on two exponential pulses

With level triggering on the raw pulses the trigger position is the later in a pulse the shorter it is. Pulses that reach just slightly above the trigger level are detected near their top, while pulses significantly stronger than the trigger level are detected near their start, as shown in Figure 19. This behaviour is called amplitude walk.

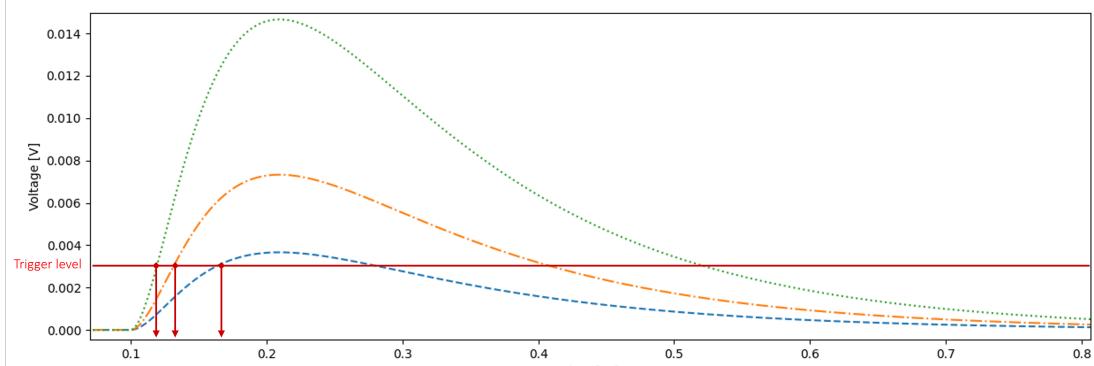
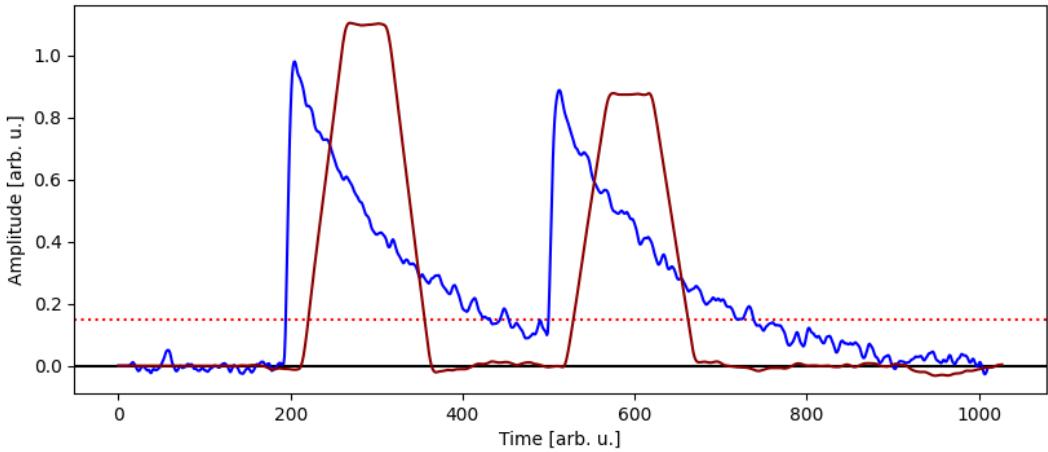


Figure 19: Amplitude walk

Due to the differential action of the boxcar filter it can be used in a digital Constant Fraction Discriminator (CFD). The point at which a pulse's derivative crosses zero after the sharp rise marks the peak of the original pulse. When using averaging from the boxcar filter this position will be slightly shifted but can still be reliably used to increase the timestamping precision.

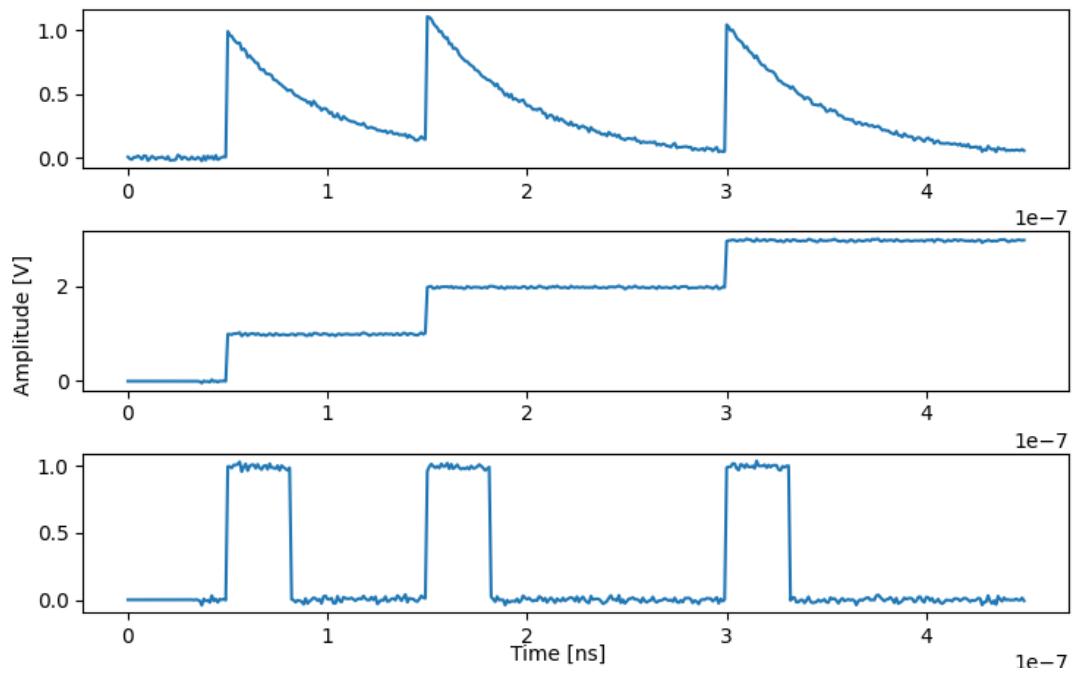
### 4.3 Trapezoidal filter

The trapezoidal filter shapes incoming pulses into the shape of an isosceles trapezoid. The sharp rising edge and slow decay tail are replaced with same length edges. Instead of a sharp peak a flattop region is formed that is proportional to the pulse height. The length of edges and the flattop can be freely controlled through constants in the filter. Figure 20 shows the effect of a trapezoidal filter being applied to a train of two exponential pulses. Note the dotted trigger line almost causing a false negative if applied to raw signal. The filtered signal provides a clear distinction between the slightly overlapping pulses.



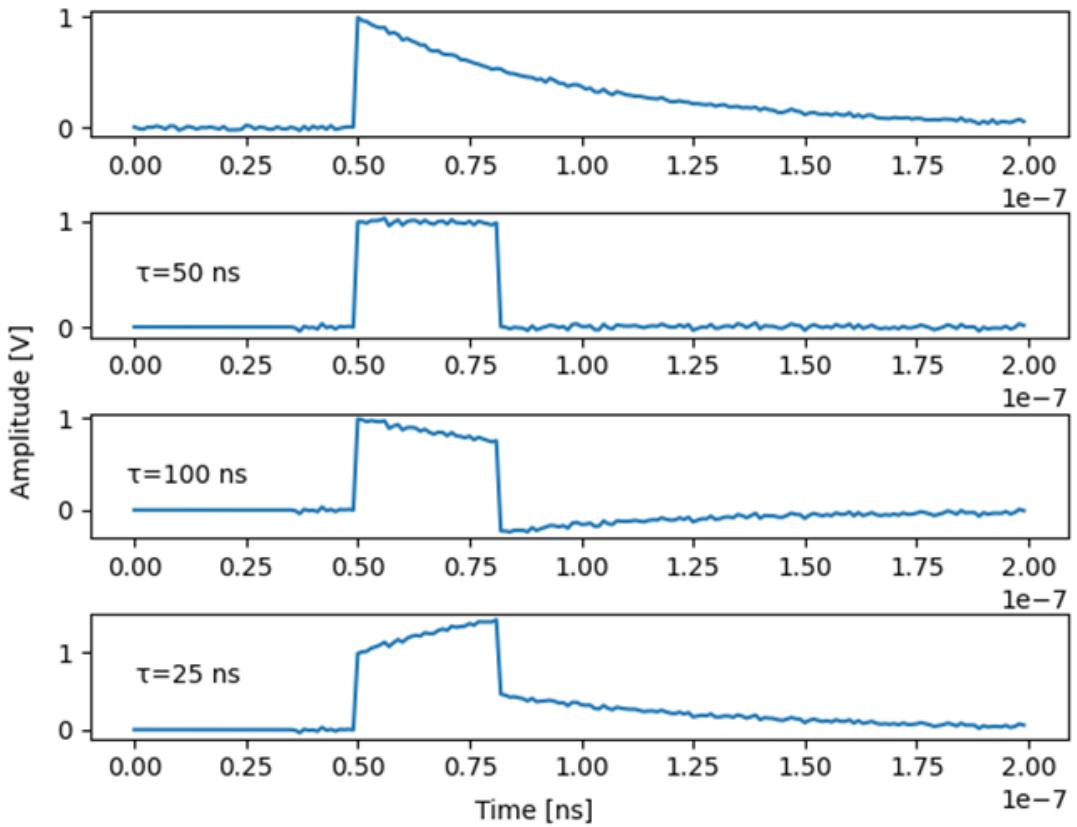
*Figure 20: Trapezoid filter applied to a train of two pulses*

The trapezoidal filter is a more complicated filter than the boxcar filter. It relies on the use of Moving Window Deconvolution (MWD), as first proposed by Georgiev et al. [39]. It is trivial to obtain a trapezoidal shape from a step input function. The MWD algorithm works to transform an exponentially decaying pulse into a step signal. A step function, while perfect for transformations, is undesirable in continuous mode of operation. With multiple pulses, the output of a MWD forms a staircase that quickly saturates. The simple solution is to perform a subtraction of a delayed sample. This changes the step signal to a rectangular shape with a length defined by the delay amount. The staircase then becomes a train of rectangular pulses. Figure 21 shows the consecutive steps of a MWD algorithm.



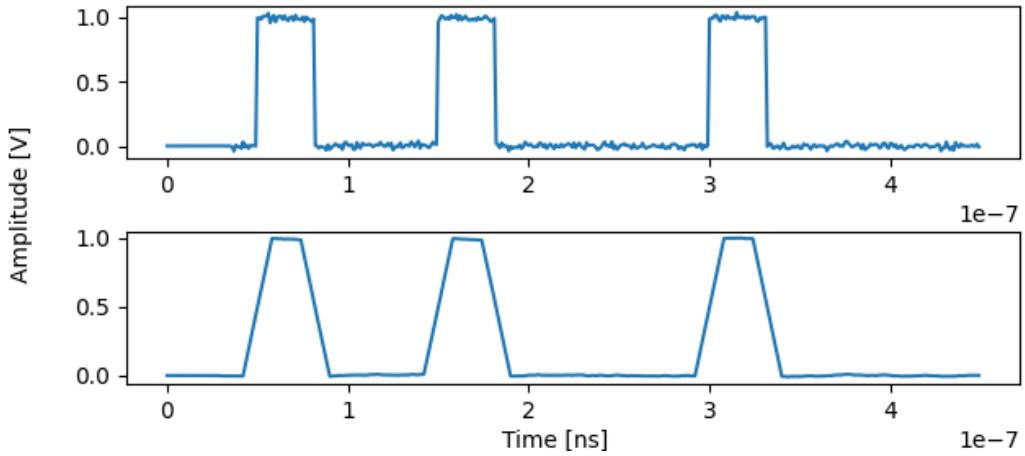
*Figure 21: MWD algorithm applied to a series of 3 pulses*

MWD requires the precise value of the pulse decay constant to be known. Figure 22 shows the effect of using a wrong value when configuring the MWD filter. By tuning the filter with a time constant that is too high an undershoot is obtained after the rectangular shape. Using a value that is too low causes an overshoot. Both mistakes reduce the accuracy of pile-up discrimination, as consecutive pulses will overlap with the decaying region and cause imprecise height measurement.



*Figure 22: The effects of decay constant mismatch between a pulse and a MWD algorithm. Decay constant of pulses is 50 ns.*

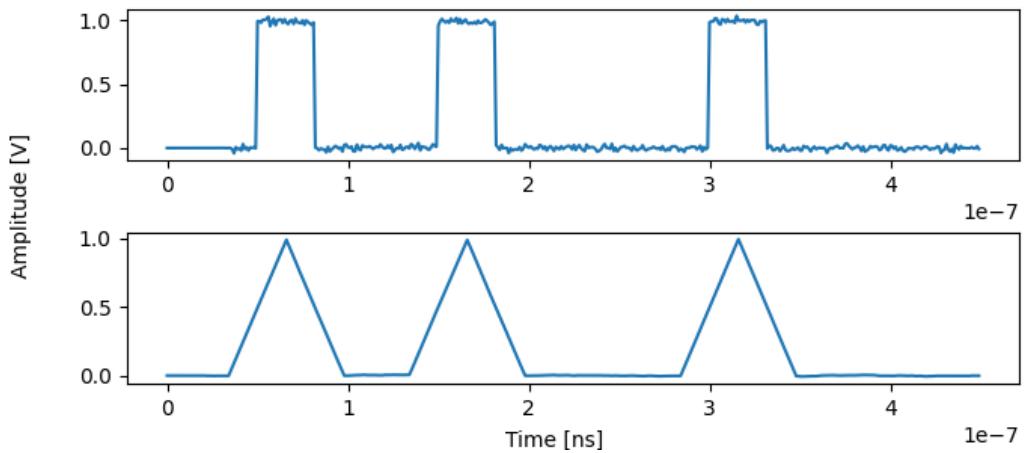
The MWD algorithm itself produces a rectangular shape. To obtain a trapezoid an averaging step is applied. This causes the edges to smoothen and as a side effect decrease the noise as shown in Figure 23. The trapezoid filter can also be synthesized using alternative methods that are better optimized for hardware implementation [40]. MWD itself requires division which is an extremely resource consuming operation in digital signal processing.



*Figure 23: Last step of synthesizing a trapezoid with the use of MWD*

#### 4.4 Triangular filter

As mentioned earlier the flattop region of a trapezoidal filter is not particularly useful for pulse detection. By using an averaging filter with the same window length as the MWD window, the flattop is squeezed together and a triangular shape is formed as shown in Figure 24. Thus, the triangular filter is a special case of the trapezoidal filter that can be used if only detection or timing is of interest.



*Figure 24: Triangular filter being formed as the last step of a MWD*

## 4.5 Other pulse detection methods

Other methods and filters for detection apart from those described above can be found in scientific literature. They have not been chosen for implementation in this work due to their low maturity, excessive complexity or subpar performance. Low hardware resource usage and minimal delay were some of the primary parameters considered. More experimental methods can provide better performance, but are harder to implement in real-time in firmware.

Faisal et al. used normalized cross-correlation to compare the similarity of incoming pulses to a predefined ideal template pulse. The algorithm was implemented in an FPGA at a sampling rate of 250 MS/s. When compared to level triggering, their method improved the probability of correct pulse detection by a factor of 4 when a template obtained from averaging capture pulses was used; and by a factor of 3 when a simplified square model was used. [41]

Kamleitner et al. analysed and compared a multitude of different edge detection algorithms. Apart from algorithms similar to the trapezoidal filter and the boxcar filter described above, the work analyzed three similarity matching algorithms, Canny's edge detector, as well as a few optimally designed FIR filters. The optimum filters won the proposed benchmarks, however, they were closely followed by the boxcar-like Single Delay Line, as well as the trapezoidal filter. Similarity filters fared significantly worse, offering superior performance only when it came to the number of false positives. [42]

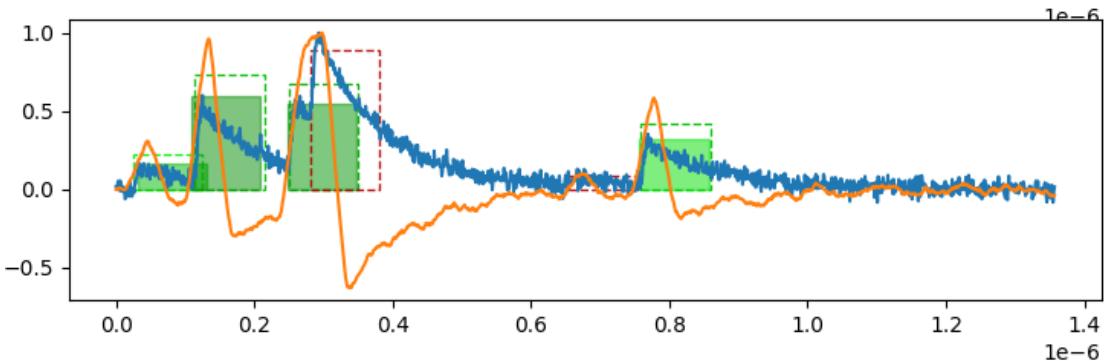
## 4.6 Simulation performance

In order to find the most suitable filters for hardware implementation in the Hard X-Ray Monitor the algorithms were first subjected to a series of simulated software tests. Level triggering, zero crossing detection, triangular and boxcar filtering were implemented with the use of the numpy and scipy libraries in Python. The same libraries were used to create a test bench for generating simulated impulses with a configurable frequency, noise, amplitude and decay constant.

The test pulse trains were fed to the filters and each output was tested with an array of different trigger configurations. The trigger level and reset at which there was the least amount of mistakes for a given algorithm was chosen as the representative record for comparison between different filters. For each record the number of properly detected pulses (true positives), missed pulses (false negatives) and noise mistaken for pulses

(false positives) were counted and divided by the total pulse count. This was done, because the pulse count was higher in tests that checked the behavior under high pile-up conditions.

For initial functionality tests and debugging, a graphical user interface was prepared with the use of the matplotlib library. The tool would run a test scenario, generate a train of pulses with the specified parameters, filter the input signal and run user-configured level triggering on the output signal. The end result would be a plot superimposing the input and output signals. Detected pulse were marked with a green rectangle. Missed pulses showed up as red dashed rectangles and badly classified noise was marked with an orange color. Figure 25 shows an example output from the graphical tool. A boxcar filter is used to improve the performance of detection. In the example one pulse is not detected due to pile-up and one is not detected due to a badly configured trigger threshold.



*Figure 25: Example output from the graphical tool for debugging detection filters*

Once the filters and the testbench were rid of bugs and optimized with the graphical tool, the code was modified for automated testing. Instead of using user-specified trigger levels the script would test a range of settings to find the best ones. In place of the graphical result a confusion matrix for each detection approach was given. The best results from each algorithm were then plotted on one graph to obtain a figure of merit based comparison for different methods.

Figure 26 and Figure 27 show the results of the simulations. The cost function used to rate the filters was a sum of false negatives and false positives divided by the total number of pulses that actually were present in the simulation. Both triangular and boxcar filters provide an immense improvement over level triggering on untreated signal. A triangular filter with the same max delay length provides slightly better accuracy than a boxcar filter.

While a filter with a longer averaging effect (longer window length) fares better in noisy environments, as indicated in figure Figure 27, the inverse is true when it comes to pile-up frequency. Shorter filters have a higher likelihood of discriminating between two overlapping pulses as shown by the results in Figure 26. There is no single detection filter that is best suited for all applications. The filter must be tuned to a specific use case, however some filters, like the boxcar offer a simpler interface for doing so, than others.

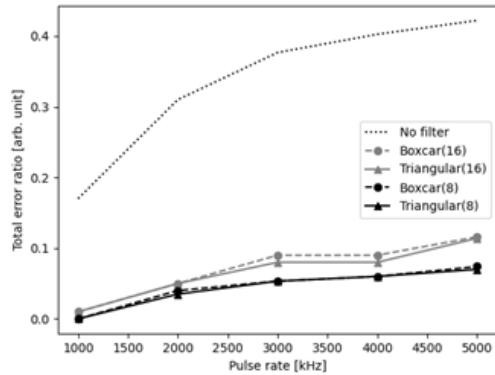


Figure 26: Detection algorithms compared under varying pulse frequency

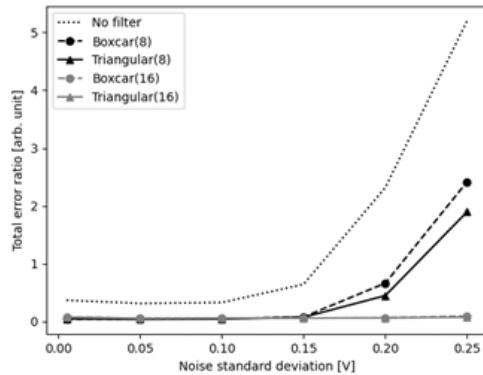


Figure 27: Detection algorithms compared under varying noise magnitude

## 5 Pulse Height Analysis

The amplitude of a pulse generated by a PhotoMultiplier Tube is directly proportional to the energy carried by a Runaway Electron. By measuring the height of the signal peak it is thus possible to estimate the RE energy and produce an X-Ray radiation spectrum. The previous chapter described methods that can be used for the detection of such pulses. This section focuses on methods that can be used on top of the detection algorithms to obtain the best possible resolution of the pulse height measurement.

### 5.1 Pulse shaping algorithms

As described in subsection 4.2 the zero crossing point of a pulse derivative pinpoints its peak. The simplest method of obtaining the pulse height is thus using a derivative and sampling the original signal at the derivative's zero crossing. The method works, however it is highly susceptible to noise. Taking only a single sample grants a result with a precision equal to the noise magnitude. It would be beneficial to take multiple samples of the signal height, to minimize the error.

Pulse shaping algorithms help reduce the influence of noise on the peak height measurement. Analogue pulse processing systems typically relied on CR-RC filters to shape pulses to a more Gaussian shape. This granted them a smoother peak that could be averaged over multiple samples. In the digital domain a trapezoidal filter (see ??) is the most commonly chosen shaping filter.

When used for pulse detection, the flattop region of a trapezoid-shaped pulse is of little interest, and can be truncated to form a triangular filter. In Pulse Height Analysis, the near constant signal produced by the trapezoid flattop is a crucial feature. Instead of taking a single sample at the sharp peak, multiple samples are taken throughout the whole flattop region. Although a longer flattop region provides better noise immunity, it cannot be increased too much due to pile-ups.

### 5.2 Integration

The pulses generated by a PMT-preamplifier combination have a constant decay time and differ in amplitude. They can be fairly accurately modelled with a single exponential formula:  $Ae^{-t}$ , where  $A$  is the pulse amplitude, and  $t$  expresses time, assuming that the

pulse peak occurs at  $t = 0$ . The surface area under the curve given by that formula is proportional to the amplitude  $A$ .

The pulses can be integrated to obtain a value that is proportional to the pulse amplitude and RE energy. As multiple samples over the pulse duration are accumulated, the effect of noise is minimized, just like with trapezoidal shaping. Integration is simpler to understand and implement than shaping and timing filters. It does not require the precise decay time to be known. This is beneficial as in the system used in this work the single-exponential function is only an approximation of the preamplified signal.

While integration only requires the use of a single accumulator to add the samples up, it results in a numerically large final sum. To obtain a usable result this value must be scaled down to a usable range. Optimally, a range corresponding to the ADC codes is chosen. An integral of a pulse is then mapped to value equal to its amplitude. The constant divider required for the mapping can be calculated for a fixed sampling window length or obtained experimentally.

The biggest drawback of using integration is its susceptibility to pile-up effects. To obtain accurate results in environments with high pile-up or long pulse decay, complicated pile-up compensation must be used. These algorithms work to subtract or otherwise remove already detected pulses from the input signal. Although such solutions have been proposed in research work, their implementation is expensive and usually limited to a single pile-up of just two pulses. [43]

### 5.3 Simulated performance

## 6 HXRM firmware

Based on the results of the research and simulations, the best timing and shaping algorithms were implemented in hardware. The ADQ14 uses Verilog and Vivado 2015.2, so these tools were chosen for the implementation of the HXRM. The developed system performs pulse detection, measures the peak height, stores the resultant spectrum and periodically transfers the results to the host PC.

### 6.1 System overview

As described in subsubsection 3.3.1 the ADQ14 DevKit grants a semi-open FPGA design with two modules that can be freely modified by the end user. The board manufacturer designed User Logic 1 with intent for it to house timing filters and User Logic 2 to contain more specialized processing logic. Initially, these ideas were followed with a Boxcar filter being implemented in User Logic 1 and a Pulse Height Analyzer being placed in User Logic 2.

With the default firmware, ADQ14 generates fixed-length records on trigger events. For example, with the window length set to a 1000 samples, upon the detection of a pulse, 1000 samples would always be transferred to the PC, regardless of pile ups and other disturbances. For this reason the timing logic originally placed in User Logic 1 was moved to User Logic 2, so that it could be better integrated with the rest of the pulse analysis systems. The UL2 module allows for full control over record lengths, so sampling windows, in which pile ups were detected, could now be rejected, split into two or combined. Figure 28 gives an overview of the system designed in User Logic 2.

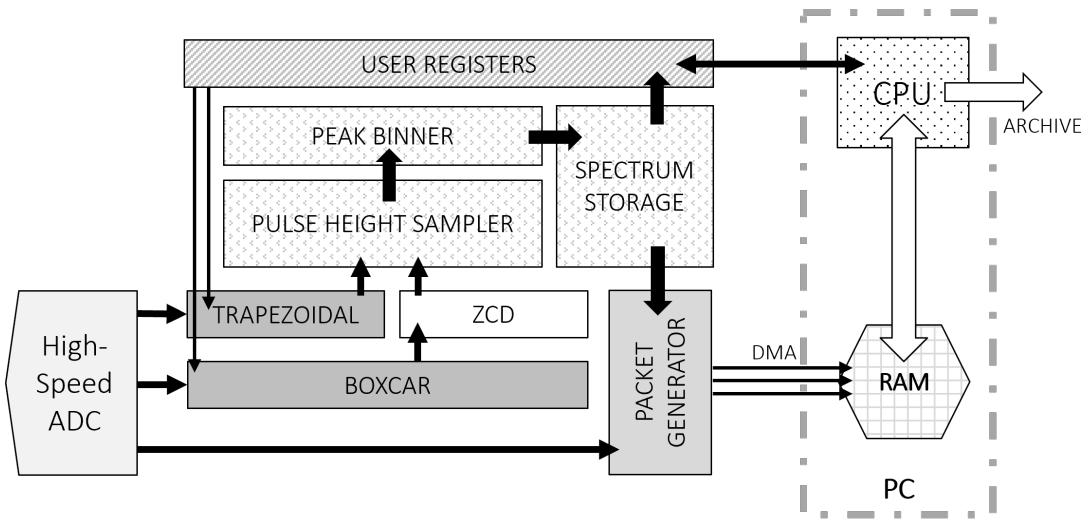


Figure 28: Firmware overview in User Logic 2

## 6.2 System control

The system is configured and controlled with the use of so called user registers.  $2^{19}$  individually addressable 32-bit words are available for reading and writing to from the PC. The first four words are reserved for internal use by the ADQ library. In the custom design, the next four have been dedicated for configuration and control as listed in Table 4. The  $2^{14}$  block of addresses starting from position 9, can be used to read the currently held spectrum.

*Table 4: User register addresses*

Address	Bits	Description
0-3	*	<i>ADQ14 Reserved</i>
4	0-31	<i>ZCD threshold level</i>
5	0-31	<i>Analog DC bias</i>
6	0	<i>PHA enable</i>
6	1	<i>PHA reset</i>
6	2	<i>ZCD trigger enable</i>
6	3	<i>Windowed spectrum enable</i>
6	4-15	<i>Reserved for future use</i>
6	16-31	<i>Record length</i>
7	0-15	<i>Spectrum window length</i>
7	28-31	<i>Spectrum bin count reduction</i>
8	*	<i>Reserved for future use</i>
9-16392	*	<i>Long window spectrum counts</i>

### 6.3 Pulse detection

The pulse detection module relies on the use of a configurable boxcar filter that then passes through a zero crossing detector. Depending on the length of the boxcar window the result of the ZCD is delayed by an appropriate amount so that it points to a point at which the trapezoidal filter's rising edge starts. To prevent false triggers the ZCD operates in two modes. First it awaits a user specified threshold value to be crossed. Only after that event occurs, does it start looking for a zero crossing. After a zero crossing the system returns back to the threshold awaiting state. Figure 29 illustrates this algorithm. The threshold value is set through user registers and does not require the FPGA to be reprogrammed to be modified.

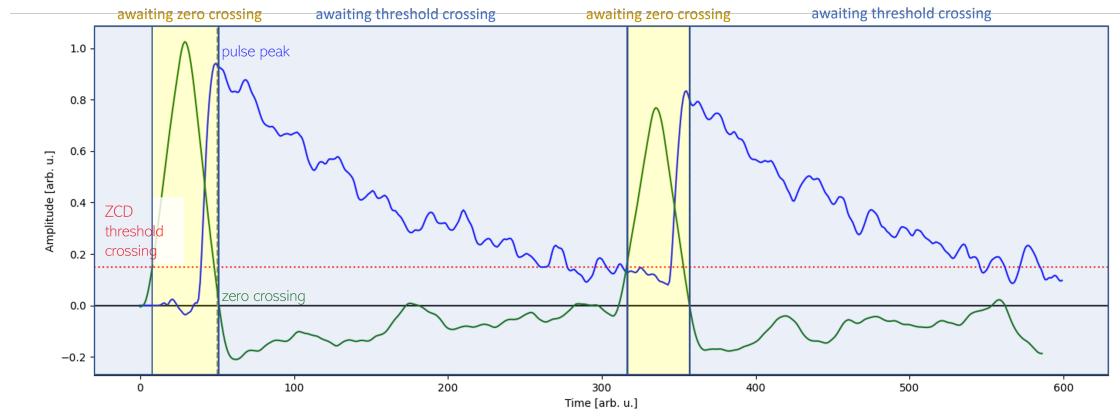


Figure 29: Zero Crossing Detection algorithm employed in firmware for pulse detection and timing

The implementation of a boxcar filter in firmware required a few considerations due to the parallel sampling described in subsubsection 3.3.2. The general formula for a boxcar filter can be described as follows:

$$y(n) = \frac{1}{W} \sum_{i=n-W+1}^n x(i) - \sum_{i=n-2W+1}^{n-W} x(i)$$

$W$  stands for the boxcar window length and  $x(n)$  is the input signal. The mathematical formula requires the use of two moving average filters. The pseudocode below describes the Register Transfer Level behaviour used to implement an optimized Moving Average filter in firmware.

```

for parsamp in [3...0]:
    ma_0[parsamp] <= x[parsamp] - x[parsamp+W];
    ma_1[3] <= ma_0[3];
    ma_1[2] <= ma_0[3] + ma_0[2];
    ma_1[1] <= ma_0[1];
    ma_1[0] <= ma_0[1] + ma_0[0];
    ma_2[3] <= ma_1[3];
    ma_2[2] <= ma_1[2];
    ma_2[1] <= ma_1[2] + ma_1[1];
    ma_2[0] <= ma_1[2] + ma_1[0];
    for parsamp in [3...0]:
        ma_3[parsamp] <= ma_3[0] + ma_2[parsamp];
        divided[parsamp] <= ma_3[parsamp] / W;
        y[parsamp] <= divided[parsamp] - divided[parsamp+W];

```

Four parallel samples are available on each clock cycle. To obtain an average they all must be added together. This requires the design to use pipelining. Attempting to add up 4 16-bit integers within a single clock cycle would violate the timing constraints specified by the FPGA. In other words, the synthesis tool can not route such a path in the device

that this addition would reliably execute at the 250 MHz clock frequency.

As the first step, a moving average is calculated for each parallel sample as if the given sample was the only sample generated on each clock cycle. One sample now outside the MA window is subtracted and a new sample is added to the current rolling sum. In the next pipeline steps the four parallel samples are added together. The proposed algorithm relies solely on two operand addition and requires one less clock cycle when compared to a simple cascading addition.

As the penultimate step a division over the window length is made. Integer division is a very costly operation in terms of hardware usage. It can, however, be avoided if the allowable window lengths are limited to natural powers of 2. In such case a division is equivalent to a right bitshift by a number equal to the power exponent. A 16-bit integer divider requires around 20 clock cycles to produce a result. A bitshift can be very efficiently performed within a single clock cycle.

Finally, to obtain the boxcar, the divided values are delayed by the window length and subtracted. Figure 30 shows the optimized accumulator being realized with DSP blocks.

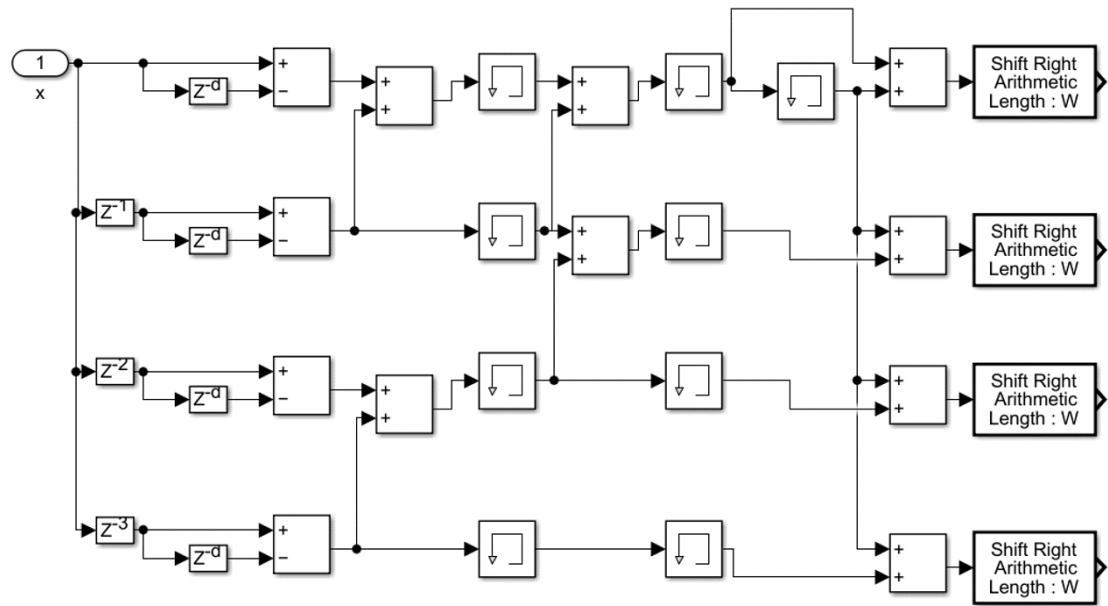


Figure 30: Timing optimized accumulator

## 6.4 Pulse height analysis

The timing generated by the ZCD is resynchronized to a trapezoidal filter, to indicate the start of the trapezoid. After a delay equal to the trapezoid edge the peak sampling module activates. The module accumulates every sample from the trapezoid flattop and divides the end result by the number of samples. The result grants an average flattop height that corresponds to the pulse peak height. The averaging action reduces overall noise and increases the measurement's precision. Figure 31 visualizes the PHA algorithm.

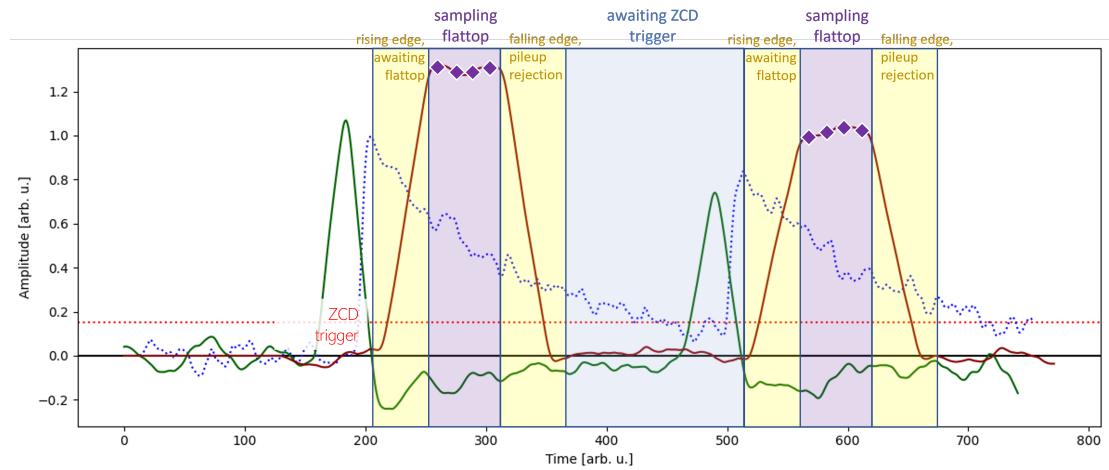


Figure 31: Trapezoid filter being used for pulse height analysis

Instead of the complicated MWD algorithm, a hardware-optimized design of the trapezoidal filter, described by Jordanov [40] is used. Figure 32 shows the filter being implemented with the use of DSP blocks. The design requires two accumulators, four addition blocks and one multiplier. Single-clock constant multiplication can be achieved with designs based on Look Up Tables (LUTs). If the operand  $M$  is a power of 2, the operation can be replaced with a left bit shift.

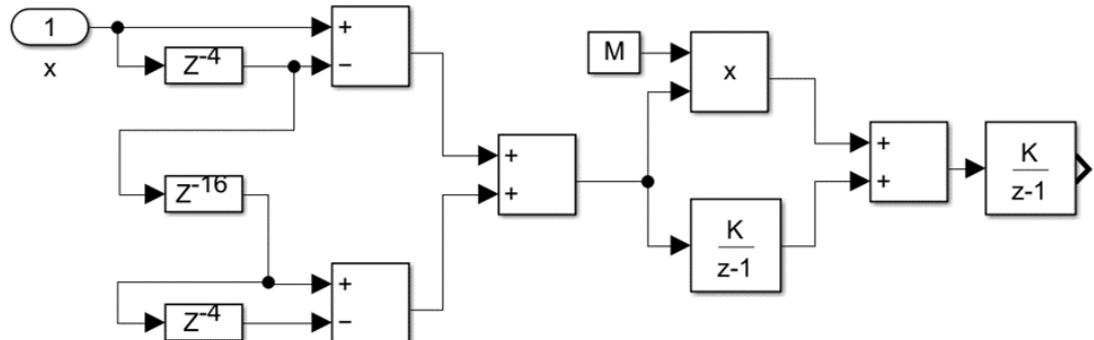


Figure 32: Trapezoid filter realized with the use of DSP blocks

#### 6.4.1 Experimental results

The custom PHA firmware was programmed onto ADQ14 and run in parallel with an acquisition process. The firmware solution performed real-time spectroscopy. At the same time the raw signal was transferred to the PC. After an hour long acquisition finished, the raw signal was processed with a software algorithm and compared with the spectrum downloaded from the FPGA.

Figure 33 shows the results of the real-time firmware spectroscopy overlaid with the offline processing method. The most important features of the  $^{137}Cs$  sample are preserved in the hardware spectrum, suggesting a successful implementation of the described algorithm. The proposed HXRM will be tested with other samples in the future.

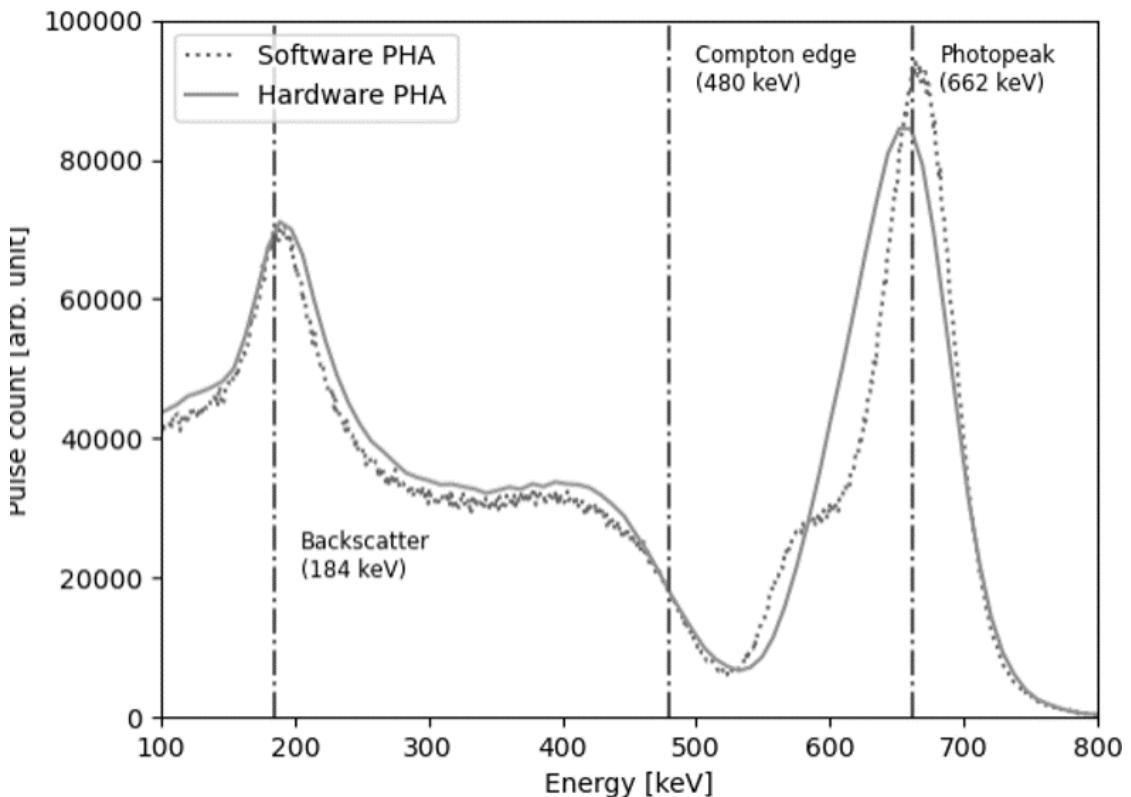


Figure 33: Spectrum obtained with the developed HXRM module

#### 6.5 Spectrum storage and transfer

The HXRM specification requires it to operate in continuous windowed mode. Spectra are collected in 10 ms windows. As soon as one window finishes, the device must start

building a second spectrum, while simultaneously transferring the first one to the PC. No dead-time should occur when starting a new sampling window.

This requirement poses a problem when it comes to the spectrum storage. With 14-bit samples the maximum theoretical resolution of the spectrum that can be obtained is also going to be 14 bits. A spectrum is essentially an array of pulse counts in each bin range. With 32 bits chosen for the count resolution a total of  $2^{14}$  32-bit words is required to store a single spectrum. This requires the use of Block Random Access Memory (BRAM) as no other FPGA memory structure can realize individually addressable increments on this scale within the timing constraints.

To prevent dead-time two BRAMs are used in an alternating mode. While one of the BRAMs is used to build the spectrum, the other one transfers its data to the PC and then resets its counts to back to zero. Once a window finishes the roles are reversed. Figure 34 shows the schematic view of this idea. For long spectrum acquisition and debugging purposes an additional third BRAM is used. This memory gets reset only on explicit user request.

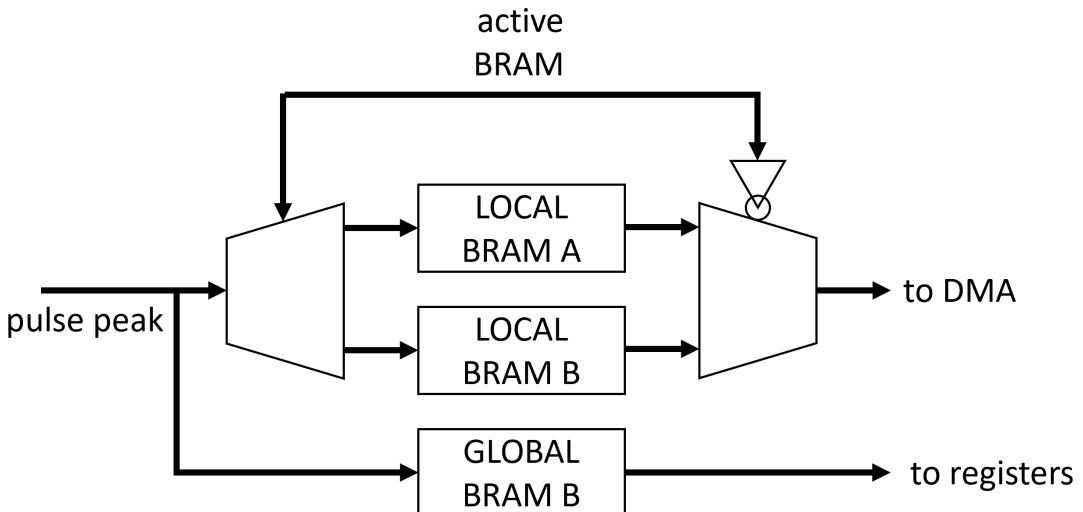


Figure 34: Spectrum storage with three BRAMs

The transfer and reset process requires the entire BRAM to be travelled twice. The ADQ14 can transfer 4 ADC samples or 64-bits of data on each FPGA clock cycle for each channel to the internal FIFO RAM. With 32-bit spectrum counts two bin counts can be transferred on each clock cycle. To achieve that both port interfaces have to be used to read two consecutive bins. This behaviour requires the BRAM to be travelled once again after transferring to reset all the count values to zero. This time both port inter-

faces write zeros to two consecutive addresses. With a 4 ns clock period the BRAM needs  $33\mu s$  to transfer its contents and another  $33\mu s$  to reset itself.

[OBRAZEK]

## 7 Data transfer

The ADQ14's datasheet specifies a maximum data rate of 3.2 GB/s with the PCIe interface. The board is sold with a PCIe 2.0 x8 interface which has a theoretical throughput of 4 GB/s. The datasheet figure is thus likely an estimate placed at 80% of the maximum or a best case scenario benchmark. In a real system the multitude of configuration options makes reaching this value complicated. ADQ14 merges all data transferred from its internal DRAM to the PC into a single stream, so the data transfer must be optimized to the highest degree. Figure 35 shows an overview of the data transfer process at different stages.

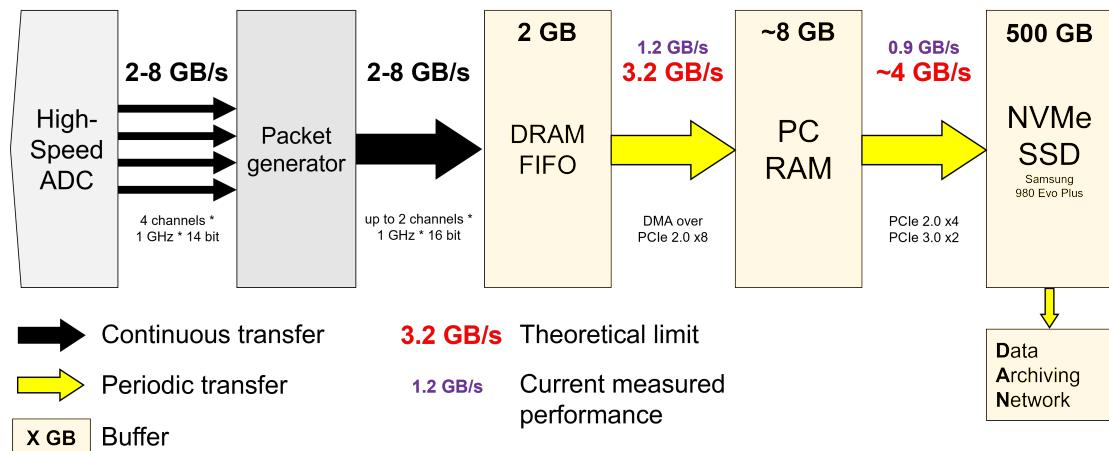


Figure 35: Data transfer in the system

### 7.1 PCIe interface

A first point that can be considered for improvement is the hardware interface itself. As of 2022 PCIe 4.0 is supported by both major CPU manufacturers. Newest Intel processors can also make use of the PCIe 5.0 standard, with AMD devices to follow suit in the near future. By upgrading the digitizer board to support PCIe 3.0 in the same x8 configuration the theoretical throughput would double to 8 GB/s. With PCIe 4.0 it would become 16 GB/s. Naturally, this is an issue that lies completely on the side of the board's manufacturer. Future digitizer boards might need to target more modern interconnection standards, to provide throughput necessary for ultra-high-speed signal acquisition.

## 7.2 Direct memory access

When using DMA, the ADQ14 splits the outgoing data into buffers. The number and size of the buffers is configurable. Depending on the desired application different settings should be preferred. To verify how the DMA buffer configuration affects transfer speed a simple test was developed. The digitizer was programmatically configured with a series of varying buffer sizes and counts and used to run 20 acquisitions, each 10 seconds long. The acquired data was not saved to disk, only copied in RAM.

Figure 36 and Figure 37 show how the average throughput changes depending on the configuration. In the tests with a varying buffer size 16 transfer buffers were used. When testing the influence of the buffers count on the throughput, the buffer size was set to 65536 bytes. Generally larger buffers, or a higher count of buffers provide better maximum throughput up to a certain maximum. Increasing the size of buffers seems to provide a more consistent improvement.

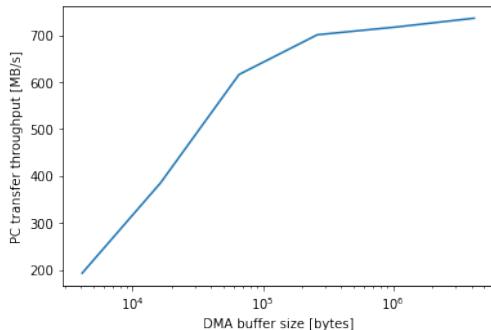


Figure 36: DMA buffer size influencing the transfer throughput

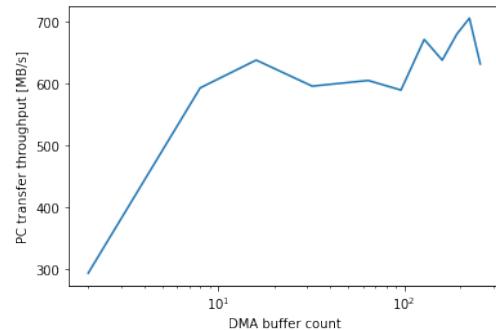


Figure 37: DMA buffer count influencing the transfer throughput

Larger buffer sizes unfortunately mean that records are packed together into larger pages. The first record of a page will be transferred with a significant delay in relation to its occurrence. For example, with a record length of 1024 and a buffer size capable of holding 10240 samples, after an event occurs, 9 more events would have to get recorded for the first one to get transferred to the PC.

ITER's specification requires a maximum of a 5 ms delay when it comes to transferring the acquired data to the Plasma Control System. If the same digitizer board will also have to transfer the raw samples for archiving and offline analysis, care will have to be taken when configuring these parameters. Transferring the raw signal will require a considerable amount of bandwidth, suggesting the need for a larger buffer size. At the same time whenever a spectrum window is finished it should be transferred to the PC

as soon as possible. Buffers no larger than a single spectrum would work best for that. Most likely a compromise will have to be reached experimentally.

### 7.3 Acquisition software

To control the acquisition process and handle the DMA data transfers, a custom software GUI application was developed. The application employs an API provided by SP Devices to interface with the digitizer board. The Qt5 framework is used for GUI display and multiple other features. Most importantly, the application leverages the Qt's signal/slot system to synchronize threaded events.

First prototypes of the application were developed with the use of PyQt, a library with Qt5 framework bindings for Python. As the scope of the application grew and the need for finer control over the performance became a priority, the application was ported over to C++ and developed further. Figure 38, Figure 39 and Figure 40 show how the application's UI changed with major releases.

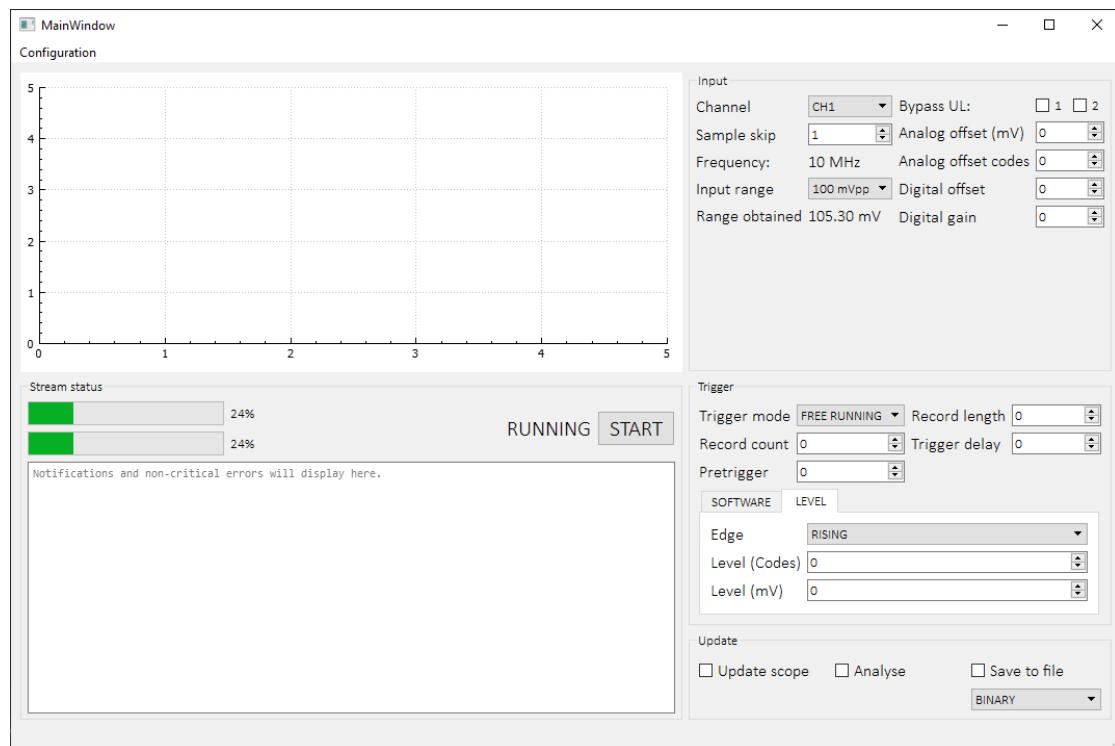


Figure 38: First functional acquisition software GUI written in C++

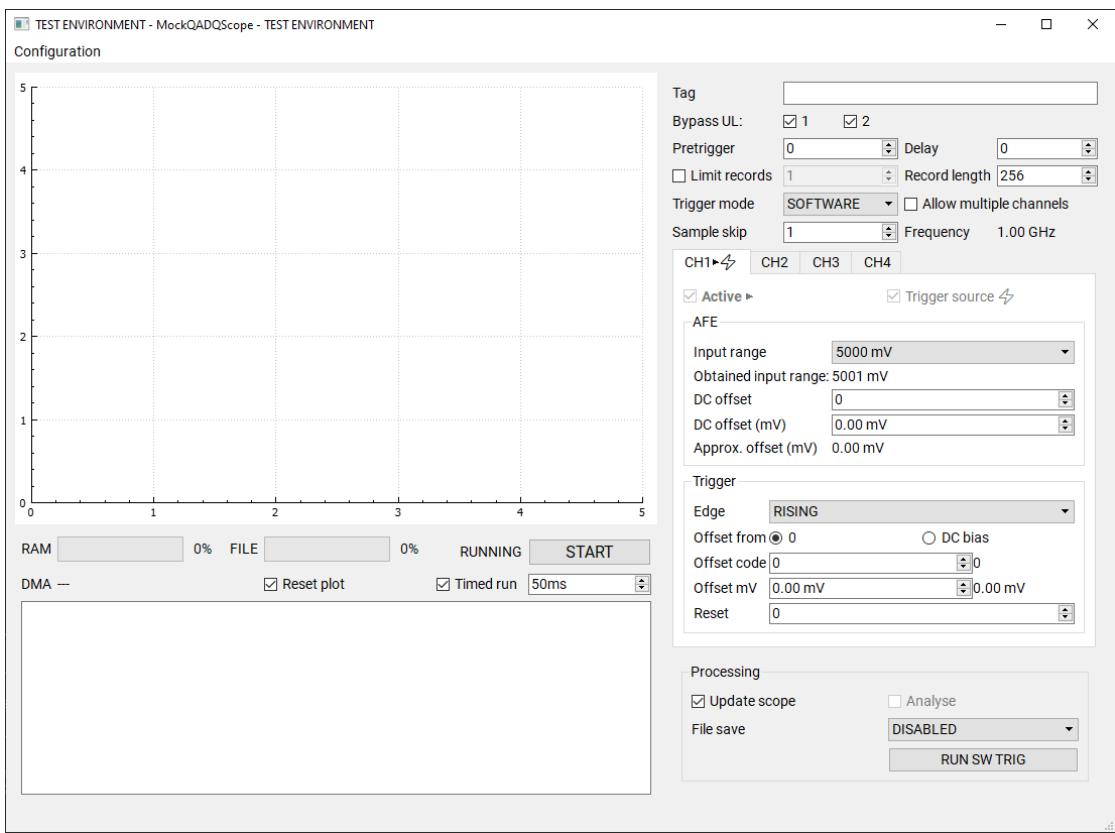


Figure 39: First major revision of the software application for acquisition

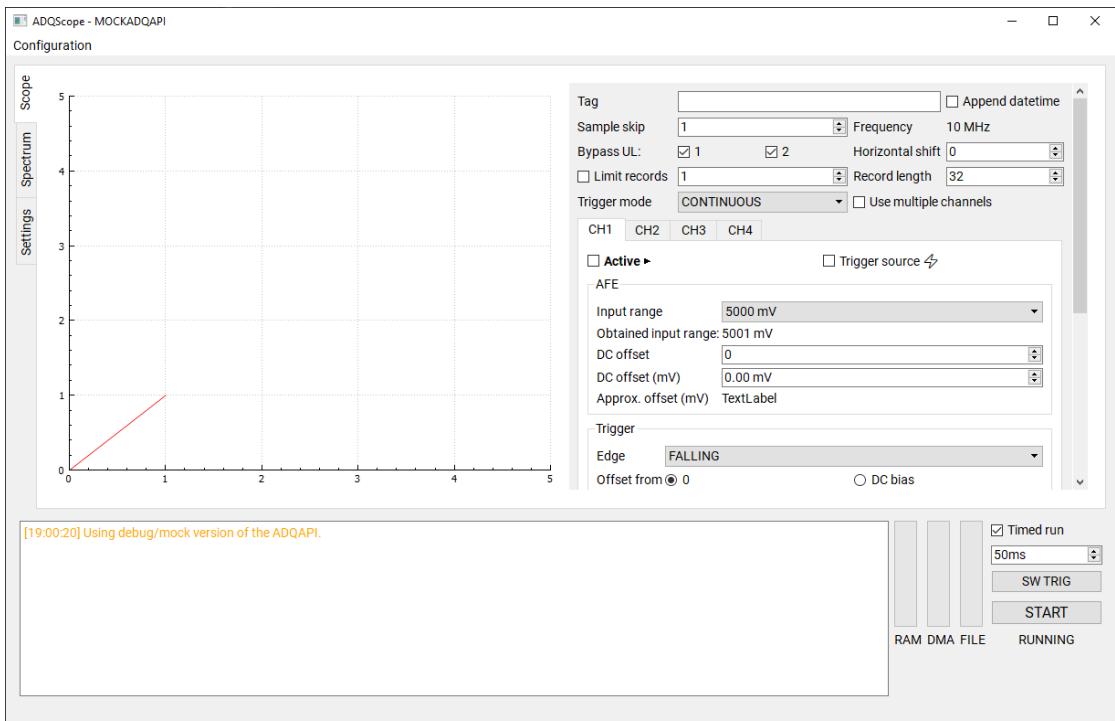


Figure 40: Current version of the software application for acquisition

### 7.3.1 Multi-threaded data transfer

The primary thread of the software application is used for UI display and general tasks like loading and saving configurations. Secondary threads are spawned to control data acquisition and archival. The proper operation of the worker threads is crucial in maintaining a high data throughput.

The acquisition application was initially developed with the use of ADQAPI library in version 55575. That version requires users to handle buffer processing and split the incoming raw data into records with a custom built coroutine. Figure 41 shows the data path in version 55575.

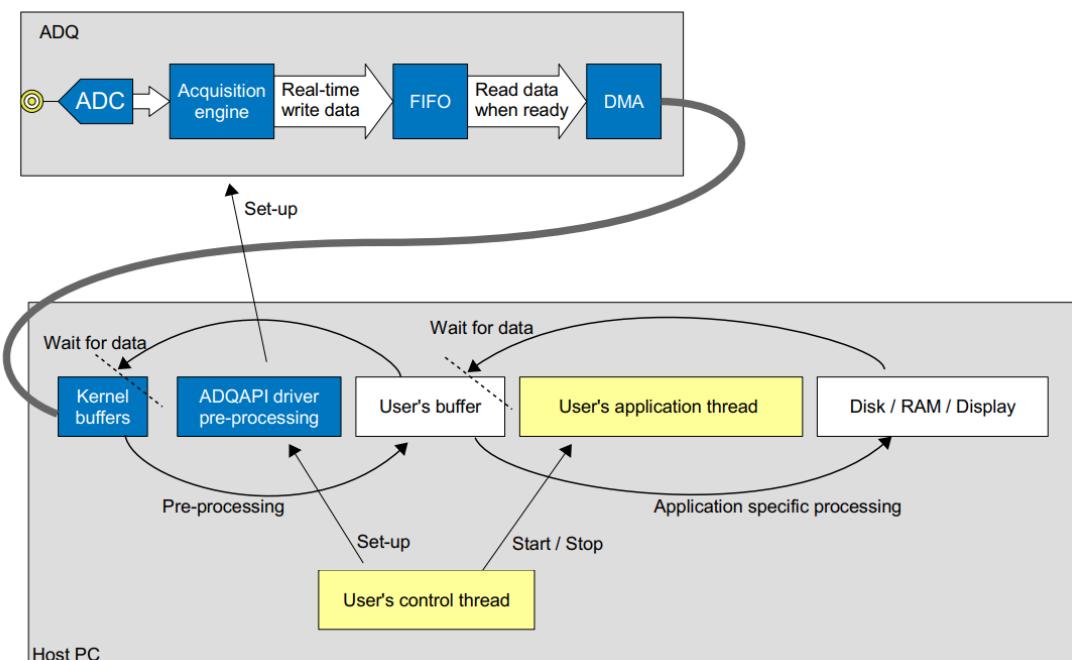


Figure 41: Buffer transfer in ADQAPI v55575[44]

Figure 42 shows the software acquisition pipeline in version 55575. The maximum size of DMA buffers is limited, so as a first step their contents is copied to another address in the PC's RAM to free their underlying memory. This is done periodically in a separate worker thread. As a DMA buffer becomes filled the worker locks one semaphore, reads its value and copies the DMA buffer to a RAM buffer pointed to by the polled semaphore value.

Another thread polls the semaphores to get the number of filled buffers. If a filled buffer is present it is read and processed. Each buffer contains a page with multiple records

and their headers (metadata). The records must be split based on the headers or a predefined configuration. Each record is then passed on to further modules. Writing records to disk is done in the same thread. If the records are to be displayed on screen they must be sent over to the primary UI thread. Once a buffer is considered processed, the appropriate semaphore is incremented to indicate that the buffer can be reused for writing.

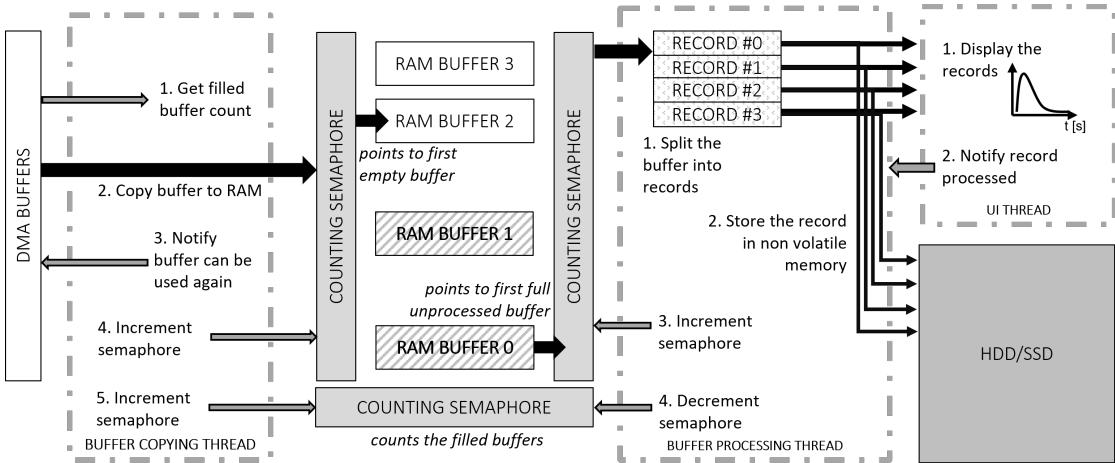


Figure 42: Threads in the acquisition app with ADQAPI v55575

Plotting signals in real time is a bottleneck in the system. With a sufficiently fast SSD drive the records can be reliably processed at a rate of up to a 1 GB/s. The display feature, relies on a third-party library QCustomPlot which requires the signed short integer samples output by ADQAPI to be converted to a QVector of QDoubles (Qt5 wrappers over std::vector and double). With as much as a few thousands of samples in each record, this is a costly operation that, combined with the thread synchronization limits the maximum throughput to around 80 MB/s. The scope is thus an optional feature intended for setup and debugging.

During the development of the application a new version of the ADQAPI library was released to customers. The primary feature of ADQAPI in version 61716, is a simplification of the buffer processing routine. The buffer copying thread is now provided by the library. Instead of operating on raw DMA buffers, users can choose to use the more abstract interface and work with already split record buffers. In testing the abstraction layer was found to offer 80-90% of the original performance, with the added benefit of removing a few bugs. The new interface was implemented in the software application going forward. Figure 43 outlines the processing routine used in the more recent version of the application.

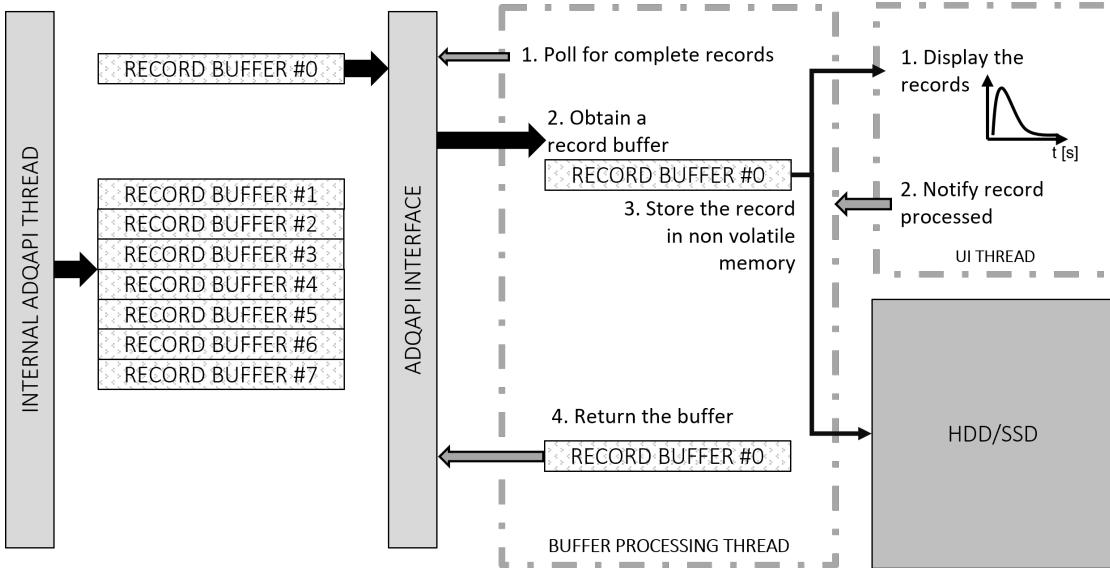


Figure 43: Threads in the acquisition app with ADQAPI v61716

### 7.3.2 Processing routine optimization

A few other improvements were made to the processing routine over time. The ADQAPI library is not thread safe. Nearly all function calls must be done from a single synchronized thread. Accessing critical features of the digitizer from two threads at once can crash the system. In initial versions of the application thread safety was achieved with a QMutex (Qt5 wrapper around std::mutex). A wrapper around the ADQAPI object would lock the mutex before every function call. This approach enabled the digitizer parameters to be accessed at any time, even during an acquisition.

With little input from the user the application would mostly perform uncontested locks, which incur only a small overhead to the function calls. The occasional contested locks could, however, potentially cause a noticeable drop in performance. To prevent that, the application was refactored to disable dynamic changes to the digitizer during an acquisition. With that done, the mutex was removed and the digitizer now exposed only a handful thread-safe methods for starting, stopping and configuring acquisitions. This change granted a small performance improvement of between 2-4% in the average throughput.

The current multithreading system relies on the use of QThreads and QObjects. The QThread class is an interface built on top of C++ threads that works together with the Qt framework event queues. In the application a worker object that inherits from QObject is moved to a secondary QThread, where it lives within an event loop. This abstraction introduces an overhead when compared to standard C++ threads, but greatly shortens

the development time. A version with a more optimized critical section is currently being developed. Its performance will be measured once finished.

### 7.3.3 Write speed optimization

Additional upgrades were made to the disk write call itself. With a large amount of calls, each writing a small buffer, a drop in performance can be observable, when compared to lower level functions. Generally, writing longer chunks of data grants better performance. A simple test was performed on the host PC to verify these facts. A series of writes was performed and their average write speed was noted. Writes were done in chunks, the size of which was changed in between tests. Figure 49 shows the results of this benchmark.

Initially the processing thread relied on calls to `std::fstream::write()`. Using `std::fstream` introduces some abstraction over direct system calls. The magnitude of the performance drop caused by write calls can differ depending on the implementation of the standard library and the computer's hardware. To get the best possible performance the application has to be built in release mode with the highest compiler optimization.

Typically writes are buffered, meaning that consecutive calls to a write function actually flush to the storage medium after some buffer is filled. Until that the content of each write is copied over to the buffer. Buffering can be disabled but in almost all use scenarios it causes a massive performance drop. Fine tuning the buffer's size to a specific hardware and use case can lead to considerable write speed gains.

To find the best approach for the processing application, a series of simple write speed benchmarks were performed for varying writing methods. `std::fstream` was used as the baseline and compared to using `FILE*` pointers with different buffer settings, including non-buffered writes. UNIX file descriptors were also tested, but caused up to an 80% drop in performance, due to the lack of buffering.

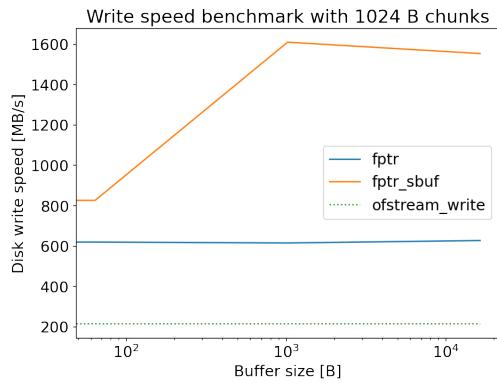


Figure 44: Write speed benchmark with a chunk size of 1024 bytes

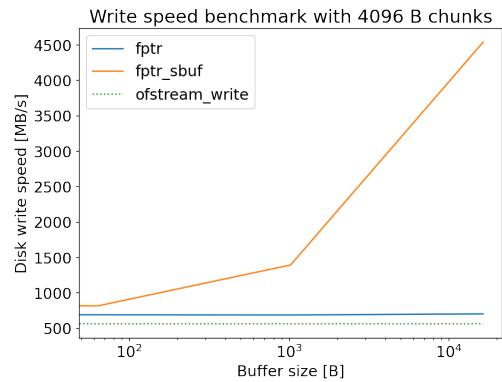


Figure 45: Write speed benchmark with a chunk size of 4096 bytes

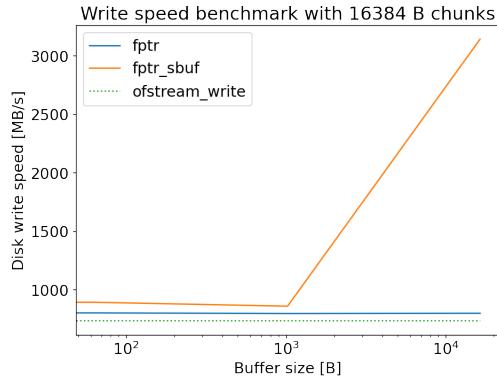


Figure 46: Write speed benchmark with a chunk size of 16384 bytes

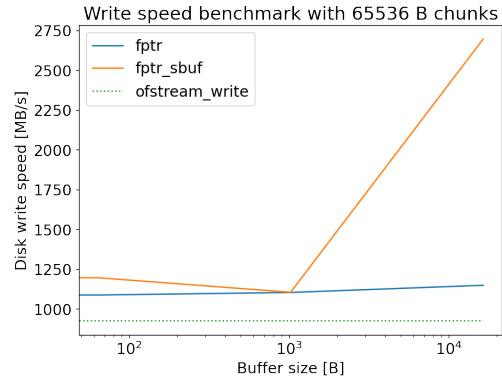


Figure 47: Write speed benchmark with a chunk size of 65536 bytes

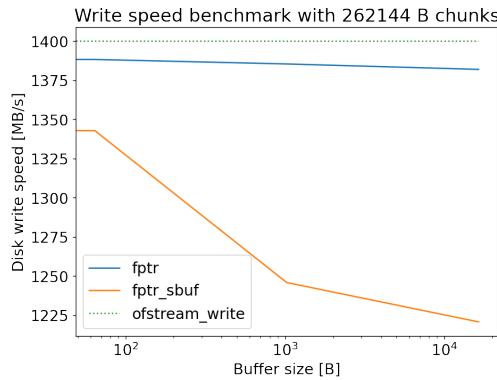


Figure 48: Write speed benchmark with a chunk size of 262144 bytes

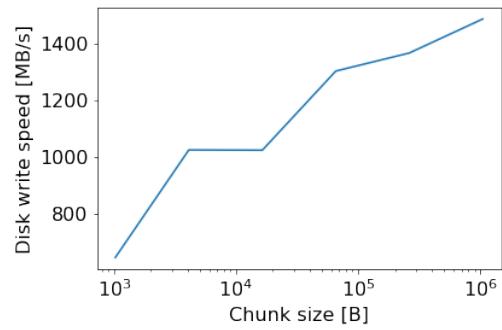


Figure 49: Average write speed as a function of the chunk size

The tests (Figure 44 - Figure 48) indicate that the highest write speeds were obtained with raw FILE\* pointers and user-controlled buffers (fptr\_sbuf) tuned to the PC's SSD. In best conditions a threefold improvement over default std::fstream configuration was

observed. Letting the standard library manage the buffer granted a stable improvement of up to 10% in average write speed. An equivalent drop was noticed when the size of the chunks exceeded the size of buffers. For the time being the application was modified to use FILE\* based calls.

## 8 Conclusions

This work described the design of a high-speed digital signal processing system for nuclear pulses based on Field Programmable Gate Arrays. Considerations and test results regarding real-time pulse detection, processing, multi-threaded acquisition systems and data transfer were presented.

A group of detection and shaping filters most commonly employed in Hard X-Ray Spectroscopy was compared. The filters were first compared in research literature and then implemented in both software and hardware to verify the results. Simulations were run with different configurations to find the best algorithm to use in plasma diagnostics. With this information the designs were implemented in a Field Programmable Gate Array and interfaced with an Analog to Digital Converter sampling with a gigasample frequency.

While some digital filters perform substantially better than others they come with different drawbacks. Fine-tuned algorithms like the trapezoidal filter offer better accuracy than simple methods like the boxcar. The cost of this increased accuracy is usually higher system complexity and lesser universality. Reduction of pile-up effects is one of the most important features of both detection and shaping algorithms.

Field Programmable Gate Arrays offer advantages over both ASICs and CPUs. In digital signal processing they offer greater parallelism than CPUs, and much better reconfigurability when compared to ASICs. This makes them a perfect choice for highly experimental systems, like tokamaks. With FPGAs changes to filters and algorithms can be introduced remotely and can involve near complete rewrites of the underlying firmware. On the other hand, although the devices allow for great performance gains, they often require complex algorithm pipelining in implementation. Additionally, the typical development time is longer than with CPUs.

Maintaining a high data throughput in digital signal processing systems is a complicated task, that requires high optimization at multiple levels. Internal buffers of the digitizer must be regularly transferred to the host PC. The host PC must be capable of processing them at speeds comparable to their appearance. This requires the use of multiprocessing. Threaded applications come with a much greater amount of risk and require a lot of optimization and debugging when it comes to their routines and synchronization mechanisms. Any software, hardware or firmware bottlenecks cause buffers to fill, which in turn greatly increase the likelihood of data loss.

## **8.1 Further problems and research**

Despite a successful implementation of the system at hand, many problems that a similar setup will have to face if used at ITER remain yet to be solved. Pile-ups still pose a significant threat to the currently used processing algorithms. Some prototypes for rejection were developed but their performance was not yet properly tested in firmware due to time and resource constraints of this work.

The tests done on the constructed HXRM setup described here involved only well-formed exponential pulses. In ITER, the light attenuation from the fiber optic interface will cause the pulse shapes to divert from exponential curves. It is possible, that entirely different detection and shaping algorithms will have to be developed for the new scenario.

Apart from the functional upgrades, other minor system improvements are planned in the near future. The critical section of the processing threads is still in the process of being optimized. The UI is currently being reworked to provide a more intuitive experience. Some changes are planned to the firmware user registers, to produce a more universal and easier to configure interface.

## References

- [1] Stuart Kirsch. „Running out? Rethinking resource depletion”. In: *The Extractive Industries and Society* 7.3 (2020), pp. 838–840. ISSN: 2214-790X. DOI: <https://doi.org/10.1016/j.exis.2020.06.002>. URL: <https://www.sciencedirect.com/science/article/pii/S2214790X20301829>.
- [2] D.H. Meadows et al. *The Limits to Growth: A Report for the Club of Rome's Project on the Predicament of Mankind.* (A Signet classic) pkt 1. Universe Books, 1972. ISBN: 9780876631652. URL: [https://books.google.pl/books?id=w%5C\\_9dAAAAIAAJ](https://books.google.pl/books?id=w%5C_9dAAAAIAAJ).
- [3] Steve Sorrell et al. „Shaping the global oil peak: A review of the evidence on field sizes, reserve growth, decline rates and depletion rates”. In: *Energy* 37.1 (2012). 7th Biennial International Workshop “Advances in Energy Studies”, pp. 709–724. ISSN: 0360-5442. DOI: <https://doi.org/10.1016/j.energy.2011.10.010>. URL: <https://www.sciencedirect.com/science/article/pii/S0360544211006694>.
- [4] IPCC et al. *Global warming of 1.5°C. An IPCC Special Report on the impacts of global warming of 1.5°C above pre-industrial levels and related global greenhouse gas emission pathways, in the context of strengthening the global response to the threat of climate change, sustainable development, and efforts to eradicate poverty.* Dec. 2018. ISBN: 9781009157957. DOI: 10.1017/9781009157940.
- [5] Alexey Mikhaylov et al. „Global climate change and greenhouse effect”. In: *Entrepreneurship and Sustainability Issues* 7.4 (2020), p. 2897.
- [6] Ibrahim Dincer. „Renewable energy and sustainable development: a crucial review”. In: *Renewable and Sustainable Energy Reviews* 4.2 (2000), pp. 157–175. ISSN: 1364-0321. DOI: [https://doi.org/10.1016/S1364-0321\(99\)00011-8](https://doi.org/10.1016/S1364-0321(99)00011-8). URL: <https://www.sciencedirect.com/science/article/pii/S1364032199000118>.
- [7] Ted Trainer. „Some problems in storing renewable energy”. In: *Energy Policy* 110 (2017), pp. 386–393. ISSN: 0301-4215. DOI: <https://doi.org/10.1016/j.enpol.2017.07.061>. URL: <https://www.sciencedirect.com/science/article/pii/S0301421517304925>.
- [8] Siddharth Suman. „Hybrid nuclear-renewable energy systems: A review”. In: *Journal of Cleaner Production* 181 (2018), pp. 166–177. ISSN: 0959-6526. DOI: <https://doi.org/10.1016/j.jclepro.2018.01.262>. URL: <https://www.sciencedirect.com/science/article/pii/S0959652618302993>.
- [9] Jack E. Fergusson. „The history of the discovery of nuclear fission”. In: *Foundations of Chemistry* 13.2 (July 2011), p. 145. ISSN: 1572-8463. DOI: 10.1007/s10698-011-9112-2. URL: <https://doi.org/10.1007/s10698-011-9112-2>.

- [10] Michael Dittmar. „Nuclear energy: Status and future limitations”. In: *Energy* 37.1 (2012). 7th Biennial International Workshop “Advances in Energy Studies”, pp. 35–40. ISSN: 0360-5442. doi: <https://doi.org/10.1016/j.energy.2011.05.040>. URL: <https://www.sciencedirect.com/science/article/pii/S0360544211003653>.
- [11] *Advanced Nuclear Reactors: Technology Overview and Current Issues*. Government Report. Congressional Research Service, 2019. URL: <https://crsreports.congress.gov/product/pdf/R/R45706>.
- [12] Steven J. Zinkle and Jeremy T. Busby. „Structural materials for fission & fusion energy”. In: *Materials Today* 12.11 (2009), pp. 12–19. ISSN: 1369-7021. doi: [https://doi.org/10.1016/S1369-7021\(09\)70294-9](https://doi.org/10.1016/S1369-7021(09)70294-9). URL: <https://www.sciencedirect.com/science/article/pii/S1369702109702949>.
- [13] Jef Ongena and Yuichi Ogawa. „Nuclear fusion: Status report and future prospects”. In: *Energy Policy* 96 (2016), pp. 770–778. ISSN: 0301-4215. doi: <https://doi.org/10.1016/j.enpol.2016.05.037>. URL: <https://www.sciencedirect.com/science/article/pii/S0301421516302658>.
- [14] Daniel Clery. „Fusion power may run out of fuel before it even gets started”. In: *Science* 376 (June 2022). doi: 10.1126/science.add5489.
- [15] Matteo Barbarino. „A brief history of nuclear fusion”. In: *Nature Physics* 16.9 (Sept. 2020), pp. 890–893. ISSN: 1745-2481. doi: 10.1038/s41567-020-0940-7. URL: <https://doi.org/10.1038/s41567-020-0940-7>.
- [16] Yuhong Xu. „A general comparison between tokamak and stellarator plasmas”. In: *Matter and Radiation at Extremes* 1.4 (2016), pp. 192–200. doi: 10.1016/j.mre.2016.07.001. eprint: <https://aip.scitation.org/doi/pdf/10.1016/j.mre.2016.07.001>. URL: <https://aip.scitation.org/doi/abs/10.1016/j.mre.2016.07.001>.
- [17] ITER. *ITER - the way to new energy*. <https://www.iter.org/>. [Online; accessed 22-July-2022].
- [18] Wendelstein 7-X | Max-Planck-Institut für Plasmaphysik. <https://www.ipp.mpg.de/w7x>. [Online; accessed 22-July-2022].
- [19] Elizabeth Gibney. „Nuclear-fusion reactor smashes energy record”. en. In: *Nature* 602.7897 (Feb. 2022), p. 371.
- [20] ITER. *On the road to ITER: milestones*. <https://www.iter.org/proj/itermilestones>. [Online; accessed 22-July-2022].
- [21] Daniel Clery and Adrian Cho. „More Delays for ITER, as Partners Balk at Costs”. en. In: *Science* 352.6286 (May 2016), pp. 636–637.
- [22] ITER. *Diagnostics*. <https://www.iter.org/mach/Diagnostics>. [Online; accessed 22-July-2022].

- [23] Pat Brans. *Archiving 20 Gigabytes per Second and Making it Usable*. <https://www.iter.org/newsline/-/3585>. [Online; accessed 22-July-2022].
- [24] ITER. *Ensuring real-time distributed computing at ITER*. <https://www.iter.org/newsline/-/3599>. [Online; accessed 22-July-2022].
- [25] Xilinx. *What is an FPGA? Field Programmable Gate Array*. <https://www.xilinx.com/products/silicon-devices/fpga/what-is-an-fpga.html>. [Online; accessed 23-July-2022].
- [26] Patryk Nowak vel Nowakowski, Dariusz Makowski, and Aleksander Mielczarek. „Low-noise Amplifier for Photomultiplier Tube Detectors for Plasma Diagnostics”. In: *2021 28th International Conference on Mixed Design of Integrated Circuits and System*. 2021, pp. 69–73. doi: 10.23919/MIXDES52406.2021.9497532.
- [27] ITER. *Generating runaway electrons in JET to benefit ITER*. <https://www.iter.org/newsline/-/2234>. [Online; accessed 22-July-2022].
- [28] Gerardo D'Elia et al. „Preliminary study of a fast massive gas injection system for plasma disruption mitigation”. In: *Journal of Vacuum Science & Technology B* 36.1 (2018), 01A103. doi: 10.1116/1.4996069. eprint: <https://doi.org/10.1116/1.4996069>. URL: <https://doi.org/10.1116/1.4996069>.
- [29] V V Plyusnin et al. „Hard X-ray bremsstrahlung of relativistic runaway electrons in JET”. In: *J. Instrum.* 14.09 (Sept. 2019), pp. C09042–C09042. doi: <https://doi.org/10.1088/1748-0221/14/09/C09042>.
- [30] M.J. Pelletier and C.C. Pelletier. „RAMAN SPECTROSCOPY | Instrumentation”. In: *Encyclopedia of Analytical Science (Second Edition)*. Ed. by Paul Worsfold, Alan Townshend, and Colin Poole. Second Edition. Oxford: Elsevier, 2005, pp. 94–104. ISBN: 978-0-12-369397-6. doi: <https://doi.org/10.1016/B0-12-369397-7/00529-X>. URL: <https://www.sciencedirect.com/science/article/pii/B012369397700529X>.
- [31] Elisabete Freitas et al. „PMT calibration of a scintillation detector using primary scintillation”. In: *Journal of Instrumentation* 10 (Feb. 2015), pp. C02039–C02039. doi: 10.1088/1748-0221/10/02/C02039.
- [32] A.G. Wright. *Why Photomultipliers Need Amplifiers*. [https://www.photonics.com/Articles/Why\\_\\_Photomultipliers\\_\\_Need\\_\\_Amplifiers/a14120](https://www.photonics.com/Articles/Why__Photomultipliers__Need__Amplifiers/a14120). [Online; accessed 23-July-2022].
- [33] G. Ripamonti, A. Pullia, and A. Geraci. „Digital vs. analogue spectroscopy: a comparative analysis”. In: *IMTC/98 Conference Proceedings. IEEE Instrumentation and Measurement Technology Conference. Where Instrumentation is Going (Cat. No.98CH36222)*. Vol. 1. 1998, 666–669 vol.1. doi: 10.1109/IMTC.1998.679876.

- [34] Esther Rani Thuraka et al. „Digital Multi-Channel analyzer for detection and analysis of radiation in nuclear spectroscopy”. In: *Materials Today: Proceedings* 38 (2021). International Conference & Exposition on Mechanical, Material and Manufacturing Technology (ICE3MT), pp. 3160–3167. ISSN: 2214-7853. doi: <https://doi.org/10.1016/j.matpr.2020.09.580>. URL: <https://www.sciencedirect.com/science/article/pii/S2214785320372965>.
- [35] Wojciech Walewski, Patryk Nowak Vel Nowakowski, and Dariusz Makowski. „Giga-sample Pulse Acquisition and Digital Processing for Photomultiplier Detectors”. In: *Journal of Fusion Energy* 41 (June 2022). doi: 10.1007/s10894-022-00320-0.
- [36] Jong-Gu Kwak et al. „KSTAR Status and Upgrade Plan Toward Fusion Reactor”. In: *IEEE Transactions on Plasma Science* 48.6 (2020), pp. 1388–1395. doi: 10.1109/TPS.2020.2964776.
- [37] R. O'Connor, J. Kadi, and S. Pandya. *System Design Description (DDD) 55.EE Hard X-Ray Monitor (H/He-Phase)*. Tech. rep. May 2021.
- [38] Patryk Nowak vel Nowakowski, Dariusz Makowski, and Wojciech Walewski. *PMT Signal Transmission for Hard X-Ray Diagnostics of Future Tokamaks*. 2022. doi: 10.21203/rs.3.rs-1529765/v1. URL: <https://doi.org/10.21203/rs.3.rs-1529765/v1>.
- [39] A. Georgiev, W. Gast, and R.M. Lieder. „An analog-to-digital conversion based on a moving window deconvolution”. In: *IEEE Transactions on Nuclear Science* 41.4 (1994), pp. 1116–1124. doi: 10.1109/23.322868.
- [40] Valentin T. Jordanov et al. „Digital techniques for real-time pulse shaping in radiation measurements”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 353.1 (1994), pp. 261–264. ISSN: 0168-9002. doi: [https://doi.org/10.1016/0168-9002\(94\)91652-7](https://doi.org/10.1016/0168-9002(94)91652-7). URL: <https://www.sciencedirect.com/science/article/pii/0168900294916527>.
- [41] Muhammad Faisal et al. „A correlation-based pulse detection technique for gamma-ray/neutron detectors”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 652.1 (2011). Symposium on Radiation Measurements and Applications (SORMA) XII 2010, pp. 479–482. ISSN: 0168-9002. DOI: <https://doi.org/10.1016/j.nima.2010.10.072>. URL: <https://www.sciencedirect.com/science/article/pii/S016890021002334X>.
- [42] J. Kamleitner et al. „Comparative analysis of digital pulse processing methods at high count rates”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 736 (2014), pp. 88–98. ISSN: 0168-9002. DOI: <https://doi.org/10.1016/j.nima.2013.10.023>. URL: <https://www.sciencedirect.com/science/article/pii/S0168900213013697>.

- [43] Mohammad-Reza Mohammadian-Behbahani and Shahyar Saramad. „A comparison study of the pile-up correction algorithms”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 951 (2020), p. 163013. ISSN: 0168-9002. doi: <https://doi.org/10.1016/j.nima.2019.163013>. URL: <https://www.sciencedirect.com/science/article/pii/S0168900219313804>.
- [44] Teledyne Signal Processing Devices Sweden AB. *ADQ14 Manual*. <https://www.spdevices.com/documents/user-guides/55-adq14-manual/file>. [Online; accessed 23-July-2022].