GPU Accelerated Microwave Kinetic Inductance Detector (MKID) Readout System

Capstone Report

Viktor Makhrinov

Nazarbayev University
Department of Electrical and Computer Engineering
School of Engineering and Digital Sciences

This project report was created on TexStudio editing platform using LaTeX. All the figures were drawn using draw.io online software tool.

NAZARBAYEV UNIVERSITY

Electrical and Computer Engineering
Nazarbayev University
http://www.nu.edu.kz

Title:
GPU Accelerated Microwave Kinetic Inductance Detector (MKID) Readout System

Theme:
MKIDs, GPU acceleration

Project Period:
Fall 2023

Project Group:
.

Participant(s):
Viktor Makhrinov

Supervisor(s):
Mehdi Shafiee

Copies: 1

Page Numbers: 24

Date of Completion:
April 26, 2024

Abstract:

Microwave Kinetic Inductance Detectors (MKIDs) have proven to be valuable instruments for detecting weak signals in the microwave and millimeter-wave spectrum ranges in many scientific domains. Their novel operating principle, which is based on the detection of kinetic inductance changes in superconducting resonators, has led to applications in astronomy, quantum computing, and materials science. The efficient reading of MKID arrays, on the other hand, creates computational challenges and frequently necessitates the use of advanced data processing techniques. While effective, traditional readout systems based on Field Programmable Gate Arrays (FPGA) have limitations in terms of flexibility and development simplicity. This study looks into the possibility of Graphics Processing Unit (GPU) acceleration to overcome these difficulties. Using an example from current research, this research demonstrates how GPU acceleration can improve MKID readout systems, improve performance, and facilitate adjustments. In addition to speeding up development, the integration of GPUs opens up novel opportunities for MKID applications across scientific disciplines.

Contents

Preface

Nazarbayev University, April 26, 2024

Viktor Makhrinov
<viktor.makhrinov@nu.edu.kz>

Chapter 1

Introduction

MKIDs, or microwave kinetic inductance detectors, are essential tools in modern scientific research. Because of their exceptional sensitivity in the microwave and millimeter-wave spectrum domains, these superconducting detectors are essential tools in many disciplines, including astrophysics, quantum computing, and materials science.

1.1 Working Principle and Applications of MKIDs

MKIDs use a simple but effective approach to monitor changes in the kinetic inductance of superconducting resonators caused by absorbed photons or particles. Photons or other particles hit with an MKID resonator, breaking Cooper pairs and increasing kinetic inductance. This shift is detectable and serves as the foundation for several applications [1].

MKIDs have been employed in astrophysical observatories, both on the ground and in space, to detect and analyze cosmic microwave background radiation, study galaxy formation, and examine dark matter [2][3][4]. Because of their high sensitivity and multiplexing capabilities, they are suitable for such accurate astronomical observations. For instance, MKIDs will be beneficials for such ground systems like the Atacama Large Millimeter/submillimeter Array (ALMA) [5]

MKIDs are appealing candidates for quantum bits (qubits) in quantum computing due to their quantum-compatible properties [6]. They promote quantum information processing by combining high-fidelity readout with low-noise operation.

MKIDs have been useful in the study of materials for determining material properties utilizing spectroscopic approaches [7][8]. Their ability to detect minor differences in the electromagnetic properties of materials has implications for material characterization and design.

1.2 Limitations of FPGA-Based Readout Systems

When reading out an array of MKIDs, a field programmable gate array (FPGA) device is often used to generate a probe signal, which is a frequency comb made up of sinusoidals. The comb is provided to the device, which is chilled in a dilution refrigerator and used to activate each resonator. When a resonator is triggered, it records its response to the relevant probe signal in the comb. On the receiving end, a filter-bank channelizer separates and analyzes each resonant frequency. This filter bank, built on the same FPGA device, consists of polyphase FIR filters [9].

MKIDs are exceptionally powerful, but their precise readout creates computing challenges. FPGAs have been widely used in conventional readout systems, but despite their exceptional capabilities, FPGAs have several intrinsic limitations:

Creating Complex Firmware: The development and modification of FPGA firmware requires a high level of competence as well as a large amount of time [10][11].

Limited Flexibility: FPGA-based systems may be unable to quickly adapt to changing experimental demands.

High Power Consumption: Warm multiplexed readout circuits can consume a lot of power, limiting their scalability for larger MKID arrays.

Cost of Expertise: Maintaining FPGA systems usually needs specialized knowledge, raising operating costs [10].

Crosstalk: High pixel density causes a crosstalk due to electromagnetic coupling between the resonators [2], and its solution also needs to be adapted for different MKIDs families, which are lumped element KIDs [12] and lens-antenna coupled distributed MKIDs [13].

Due to these limitations, there is a rising demand for investigating alternate readout techniques that combine computational effectiveness with development simplicity.

Recent study employed the MGTree technique which is known for its remarkable performance in virtual source tree tracing for 3D ray tracing applications. It accomplishes this by innovatively combining CPU, GPU, and CUDA technologies in a single system, just the one used in an exemplary study for this paper.. Their approach, which uses MPI and CUDA, effectively utilizes the parallel processing powers of both CPU and GPU, making it excellent for multi-operand calculations, global performance optimization, and job distribution. Notably, CUDA's connection to the GPU helps to reduce the burden on individual cores, ensuring scalability and optimal resource utilization. MGTree proves its worth in real-world ap-

plications by outperforming traditional CPU and GPU methods, such as wireless channel simulations in an indoor conference room. This emphasizes the benefits of integrating CPU, GPU, and CUDA into a unified computing framework[14].

This paper aims to show that using GPUs gives a compelling potential to revolutionize MKID reading in light of the operating principles and applications of MKIDs, their limitations in conventional FPGA-based readout systems, and the benefits of GPU acceleration. It will be discussed, how MKID readout systems can solve present challenges and open up new opportunities by utilizing the parallel computing power, high memory bandwidth, scalability, accessibility, and ease of programming inherent in GPUs. The use of GPU acceleration in MKID readout will be discussed in greater detail, along with how it affects performance and adaptability as well as the potential for revolutionizing a number of scientific fields. We want to show how GPU-accelerated MKID readout systems may enhance readout responsiveness and efficiency, expanding the possibilities of MKID detectors across several scientific study disciplines.

Chapter 2

Background

2.1 Field-Division Multiplexing (FDM) in MKID Readout

It is important to fully understand the core concepts of MKID readout, in particular the function of Field-Division Multiplexing (FDM), before getting into the benefits of GPU acceleration. In MKID readout systems, the FDM technique is used to separate the signals coming from different resonators in an array. In an MKID array, each resonator reacts to a certain frequency. Multiple resonators can be read out simultaneously by using FDM, which divides the resonators into discrete frequency bins [15][16]. The MKID readout algorithm typically goes through three stages:

Generating Probe Signals at Low Frequencies: To read an array of N microwave resonators, N different probe signals are initially produced, each at a different frequency. In order to increase the frequency of these signals to the required level, which coincides with the frequency of the resonators, I/Q modulation is used initially to generate them at a low frequency [16][17].

I/Q Modulation: Up-converting a signal to a higher frequency involves using the I/Q modulation technique. A local oscillator (LO) signal at the desired frequency is added to the signal to obtain the desired outcome. The mixer produces a signal at the sum and difference of the two input frequencies [16][17].

Excitation of the MKIDs: After generation, the probe signals are given to the MKIDs in order to excite them. In response, the MKIDs absorb photons and modify their resonance frequency. By keeping an eye on the phase and amplitude of the reflected probe signals, it is possible to track changes in resonant frequency [16][17].

Due to the need for effective readout systems given the process' complexity, MKID technology is a prime candidate for GPU acceleration

2.2 Advantages of GPU Acceleration for MKID Readout

Parallel Processing Power The immense parallel processing capacity of GPUs is one of their most notable characteristics. GPUs are made for performing several parallel activities at once, in contrast to Central Processing Units (CPUs)[18], which are intended for sequential processing. This architecture perfectly satisfies the demands of MKID readout, where handling massive datasets over numerous channels is crucial [19]. MKID detectors frequently consist of arrays with hundreds or even thousands of different resonators, each of which generates data that needs to be analyzed simultaneously. Such data-intensive operations are well suited for GPUs, which makes them a suitable solution for MKID readout applications [20][21].

High Memory Bandwidth The larger memory bandwidth of GPUs is another significant benefit. As a result, data transfer bottlenecks are reduced due to the GPUs' ability to efficiently manage and transmit data between processing cores and memory. High memory bandwidth ensures that the GPU has the ability to modify the data with the bare minimum of delays in MKID readout, where quick data capture and processing are essential. This results in quicker reading times and better system performance overall [20][22].

Scalability The scalability provided by GPUs fits the many requirements of MKID studies. GPUs can adjust to the computing demands, whether working with small-scale systems or huge detector arrays. This scalability is especially advantageous because it enables researchers to customize their readout systems to the precise needs of their investigations. Additionally, it guarantees that GPU-accelerated MKID reading systems can expand along with changing initiatives for research [20].

Accessibility and Ease of Programming GPU-based systems are more widely available and easier to program than FPGA-based ones. Since FPGA creation often involves specific education and expertise, the number of professionals qualified to update or maintain the firmware is constrained. GPUs, however, can be programmed using well-known languages and libraries. For instance, the CUDA platform from Nvidia provides a C++-compatible framework for GPU programming [19]. By making MKID technology more accessible, a larger group of scientists and researchers are able to take advantage of its benefits [10][22].

2.3 Example from the Field

Examining a case from recent research, as described in source [10], will help to demonstrate the vital potential of GPU acceleration in MKID readout. The ROACH (Reconfigurable Open Architecture Computing Hardware) boards from the Casper

collaboration, which typically rely on FPGA firmware for reading, were enhanced with GPU acceleration in this study. The original purpose of the ROACH boards, which use Xilinx FPGAs, was to produce buffers for digital-to-analog converter chips (DACs) and to process raw data obtained by analog-to-digital converter chips (ADCs) in real-time. The substantial knowledge required for firmware development and modification in this FPGA-centric strategy resulted in lengthy development timeframes.

GPU acceleration was implemented in the modified system to shift computationally heavy activities from the FPGAs to the desktop computer's GPUs. The development time for the required software, which was built in Python and C++, was substantially cut down to only a few days by utilizing Nvidia's CUDA framework. Researchers were able to quickly modify the readout system to satisfy the demands of various detector programs because of this design approach. The addition of GPUs sped up development while also improving the system's adaptability and flexibility. As the needs of the experiments changed, researchers were able to quickly modify the readout algorithms. This effective use of GPU acceleration shows how capable and quick-acting MKID reading systems could potentially be due to this technology.

Chapter 3

Methodology

3.1 Readout algorithm in the exemplary study

Exemplary study used in this work is a paper “A Flexible GPU-Accelerated Radio-frequency Readout for Superconducting Detectors” by Lorenzo Minutolo et. al. from 2019 [10]. To quickly summarize, in this research the Casper team included GPU acceleration into ROACH boards. Historically, these devices relied on FPGA firmware for data reading. The first tasks of Xilinx FPGA-equipped ROACH boards were real-time processing of raw data from ADCs and buffer construction for DACs. Committees were initially responsible for implementing these standards. Due to the skill set needed to build and alter firmware using this FPGA-centric strategy, development took longer. The revised system moved computationally heavy activities from FPGAs to desktop GPUs using GPU acceleration technology. Implementing Nvidia’s CUDA library cut Python and C++ applications’ development time to days. Due to rapid development, scholars may quickly adjust the readout system to meet detector program parameters. GPUs accelerated development and increased the system’s adaptability and versatility. The researchers showed they could quickly modify readout algorithms to change experimental needs. GPU acceleration has improved MKID reading systems’ efficiency and responsiveness.

All of the codes for the GPU server and the readout used in this paper ca be accessed via GitHub page titled ”GPU-SDR” and these will be used In order to investigate their methods and steps they take for the readout algorithm. As an example for the measurement and analysis I take the “swipe-parameters.py” script py which is according to the author "the most similar to our acquisition routine". This scripts employs a comprehensive methodology to examine the functionalities of the Vector Network Analyzer (VNA) in meticulous concentration, with respect to the readout of Microwave Kinetic Inductance Detectors (MKIDs). The modular architecture of the system enables a thorough examination of the critical phases

that are paramount in the overall assessment of the experimental configuration. At each stage of the algorithm, its importance and contribution to the overarching understanding of the system are emphasized. Methodically, the algorithm is developed.

To initiate the process, a multiplication of a seed VNA measurement is performed to guarantee the preservation of phase coherence. A rise in processing efficiency can be observed as a consequence of employing GPU acceleration during server initialization. Post-processing line delay measurement and analysis is crucial in order to effectively address the inherent delays that are present within the system. For subsequent measurements to be conducted with greater precision and accuracy, the system's complexities must be thoroughly examined. The seed VNA measurement is subsequently performed by the script utilizing the Single-VNA function, which is tasked with acquiring comprehensive frequency sweep data. Crucial resonance characteristics are acquired during this phase, which is crucial for gaining a comprehensive comprehension of the behavior of MKID resonators. The subsequent analysis encompasses the procedures of initializing, fitting, and exhibiting the resonator. By employing this approach, crucial parameters that are required for subsequent iterations of the analysis can be extracted. The program executes VNA measurements through a smooth transition to a loop that traverses a range of gain configurations. Preliminary resonator initialization is accomplished by employing the most recent VNA scan or the seed VNA. This is an essential stage in order to maintain the coherence of the experimental configuration. Methodically fitting resonators is performed so as to acquire information that is beneficial with regard to the way in which the instruments react to different levels of gain. In order to ascertain the dynamic behavior of MKID resonators under various operational conditions, this step is necessarily undertaken. Acquisition of ambient data and meticulous collection of tones during each iteration constitute a critical component of the script. This phase offers a thorough comprehension of the noise properties inherent in the system, which is significant for applications in condensed matter physics and quantum technologies. Capitalizing on this element is of utmost importance. An enhanced flexibility of the script is achieved through the transition of the resonator group from the VNA to the noise files, enabling a comparative analysis of the resonator's response in the presence and absence of external effects. The diagnostic VNA noise graphs, which are generated at the conclusion of each workflow iteration, offer a graphical depiction of the noise characteristics. During this stage, not only is the prompt evaluation of the experiment's results facilitated, but the experimental setup's characteristics are also ascertained and improved. The script streamlines critical processes within the framework of MKID analysis by leveraging GPU acceleration at critical phases via the Vector Network Analyzer (VNA). This results in improved computational efficiency and accelerated operations. GPU acceleration, which is enabled on the server through

parallel processing, provides significant advantages when complex computations and procedures involving large amounts of data must be executed more rapidly. Procuring a seed VNA measurement, which serves as the preliminary phase of the script, is an operation that requires a significant quantity of CPU resources. A GPU acceleration mechanism may be implemented during the server's launch in order to maximize the utilization of the GPUs' parallel processing capabilities. As a result, a substantial reduction in time is anticipated for the coherent replication of phase-sensitive measurements. The accelerated execution will ensure that the ensuing operations, which rely on precise and prompt initialization, commence with a strong foundation. During the second stage of measuring and evaluating line delay, which is essential for compensating for delays introduced by the system, the graphics processing unit's (GPU) acceleration is maximally significant. GPUs employ parallel processing capabilities in order to expedite delay computation process. This allows for expeditious modifications and enhances the overall accuracy of the system. While the Single-VNA function is being executed for seed VNA measurement, GPU acceleration is implemented to accelerate the process of acquiring frequency sweep data. It is critical to possess this acceleration when handling a substantial quantity of data points and iterations. The effective acquisition of detailed resonator parameters is facilitated by the parallel processing capabilities of GPUs, thereby enhancing the overall pace of the workflow.

3.2 Accelerating MKID Readout with GPU

The script streamlines critical processes within the framework of MKID analysis by leveraging GPU acceleration at critical phases via the Vector Network Analyzer (VNA). This results in improved computational efficiency and accelerated operations. GPU acceleration, which is enabled on the server through parallel processing, provides significant advantages when complex computations and procedures involving large amounts of data must be executed more rapidly.
Procuring a seed VNA measurement, which serves as the preliminary phase of the script, is an operation that requires a significant quantity of CPU resources. A GPU acceleration mechanism may be implemented during the server's launch in order to maximize the utilization of the GPUs' parallel processing capabilities. As a result, a substantial reduction in time is anticipated for the coherent replication of phase-sensitive measurements. The accelerated execution will ensure that the ensuing operations, which rely on precise and prompt initialization, commence with a strong foundation.
During the second stage of measuring and evaluating line delay, which is essential for compensating for delays introduced by the system, the graphics processing unit's (GPU) acceleration is maximally significant. GPUs employ parallel process-

ing capabilities in order to expedite delay computation process. This allows for expeditious modifications and enhances the overall accuracy of the system.
While the Single-VNA function is being executed for seed VNA measurement, GPU acceleration is implemented to accelerate the process of acquiring frequency sweep data. It is critical to possess this acceleration when handling a substantial quantity of data points and iterations. The effective acquisition of detailed resonator parameters is facilitated by the parallel processing capabilities of GPUs, thereby enhancing the overall pace of the workflow.

A procedure frequently employed in signal processing, data reduction, and large-scale physics simulations [23] is the FFT. In certain scientific and professional fields, the Fast Fourier Transform (FFT) is a crucial operation. In addition, a substantial quantity of resources and memory bandwidth are required to support the computational demands of this method, specifically for extended Fast Fourier Transforms (FFTs).
The potential performance enhancement of applications demanding substantial computational capacity has been demonstrated to be tenfold when GPUs are utilized in contrast to conventional multi-core CPUs[23]. This situation offers a potentially fruitful prospect to tackle these specific obstacles. Reducing the challenge of fitting FFT problems within the GPU's memory and enhancing the data transmissions between the CPU and the GPU constitute the majority of the current efforts to implement GPUs in FFT applications.
Implementing strategies such as decomposing Fast Fourier Transform (FFT) problems and optimizing the architecture of processing elements in GPUs [23] has been crucial in making use of the hierarchical memory structure inherent in GPUs. By utilizing the Fast Fourier Transform (FFT) method, it is feasible to divisible a Discrete Fourier Transform (DFT) into more manageable components. Numerous implementations, including the Cooley-Tukey Fast Fourier Transform, decrease the computational expense of the algorithm from O(N2) to O(N log N).
In the realm of radio astronomy data correlation, the ability to identify distinct signals in the frequency domain is a critical component of the scientific discipline. In this regard, FFT is an indispensable instrument. The Fastest Fourier Transform in the West (FFTW) and the CUDA Fast Fourier Transform (CUFFT) serve as examples of how GPU acceleration is implemented to enhance the efficacy of Fourier transform calculations.
GPUs have become essential components in parallel computing applications, with cosmology being one such specific domain [24]. GPUs are widely recognized for their robust computational capabilities and intrinsic parallelism and they are distinguished by a multiplicity of data parallel threads, operate under a programming paradigm that is diametrically opposed to that of conventional CPUs. Originating as parallel accelerators for scientific computation, GPUs have undergone a sig-

nificant transformation since their initial purpose of powering visual applications [25]. They now outperform CPUs in terms of processing capabilities. Considerable scholarly inquiry has been devoted to GPU parallelization techniques for FFT-related applications in the field of radio astronomy correlation. In pursuit of optimizing the functionality of the GPU, a variety of strategies are implemented. These approaches consist of single-threaded CPU methods and highly parallelized pair parallel strategies. These solutions, which consider GPU memory operations, data staging, and the intricate nature of thread balancing [25], serve as illustrations of the intricacy that is intrinsic to the optimization process.

3.3 Replicate measurements from exemplary study

3.3.1 Experimental setup and system architecture

Our project's major goal is to recreate and implement a system that Minutolo[10] first presented in the paper mentioned above. To acquire information regarding incoming radiation, the system uses Microwave Kinetic Inductance Detectors in conjunction with a Software Defined Radio configuration. The USRP X310 acts as the system's primary component, and it is equipped with a UBX160 daughterboard. It is designed to provide a bandwidth of 100 Msps and the ability to manage frequencies up to 6 GHz.

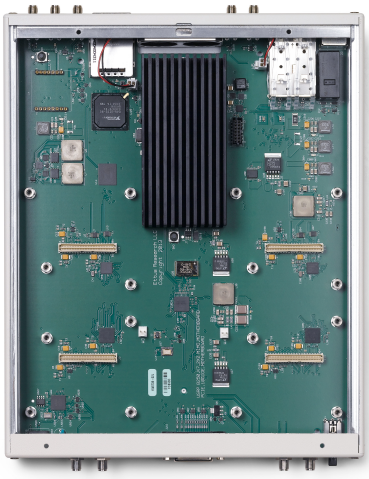

Figure 3.1: USRPx310 FPGA[26]

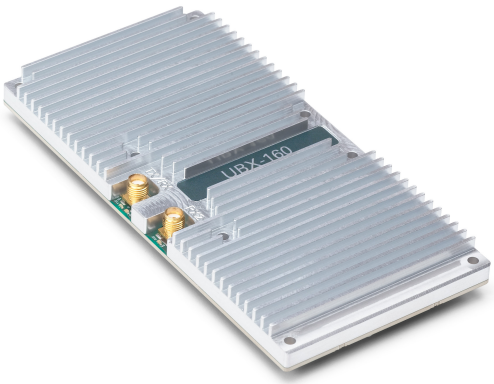

Figure 3.2: UBX160 daughterboard for USRP[27]

The provided block diagram explains in detail the signal processing architec-

ture with which the USRP X310 was designed in terms of hardware. This is accomplished through the use of an intricate layout in which the transmitter path enhances the outgoing signals, which are then transferred via a TX/RX antenna. In contrast, the receiver path collects incoming signals using a separate RX2 antenna. Following that, these signals are amplified and filtered using low-pass and band-specific filters to accommodate different frequency ranges. The complex conversion paths are particularly significant; the down-conversion path uses a reference signal to ensure precision, effectively integrating high-frequency signals into a lower intermediate frequency or directly into baseband I/Q components. Before the signal is subjected to the final I/Q modulation for transmission, the up-conversion circuit includes a 2.44 GHz band-pass filter. This filter helps to improve the signal's clarity. This strong architecture, which combines high-speed data converters and FPGA-based signal processing, enables the transfer of high-rate data, which is required for the exact operation of MKID detectors.

Another advantage of our system is its high computing power. The SDR system

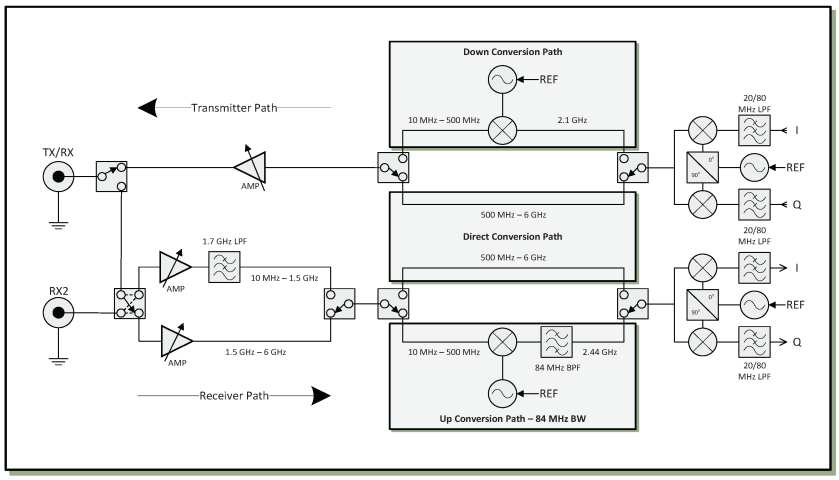

Figure 3.3: Block diagram of UBX160 daughterboard[27]

uses a 10-gigabit network card with an SPF+ cable connector to transport down-sampled data to a GPU server. The server is equipped with a Tesla K40m graphics processing unit, 140 gigabytes of random access memory (RAM), and a 64-core processor, all of which give enough of computing capacity to handle the high amount of data flow produced by MKIDs.

3.3.2 Performing measurements

A Python-based program deployed on the GPU server performs several analytical operations on the input signals. This algorithm's processing includes techniques

such as noise acquisition, delay measurements, resonator calculations, and analysis with a Vector Network Analyzer. When the readout begins, the GPU server is started, which is given by the GPU SDR repository. This server is launched using the GCC compiler, which connects to the Boost and UHD libraries, which are then used to configure the USRP and make measurements on the FPGA. In addition, a connection is formed between the server and the graphics processing unit (GPU), in this case the Tesla k40m. The GPU is used to handle a large amount of data that is transmitted via USRP in the form of MKIDS. While it is feasible to start the GPU server from the computer and send measurement commands remotely, our configuration requires that all operations be performed within a single setup. This is done to improve dependability and reduce data transmission delay by reducing the number of connected devices. Finally, the server connects to the USRP and the whole system is ready to perform measurements.

A second Python session is used to assist the sending of orders for signal analysis.

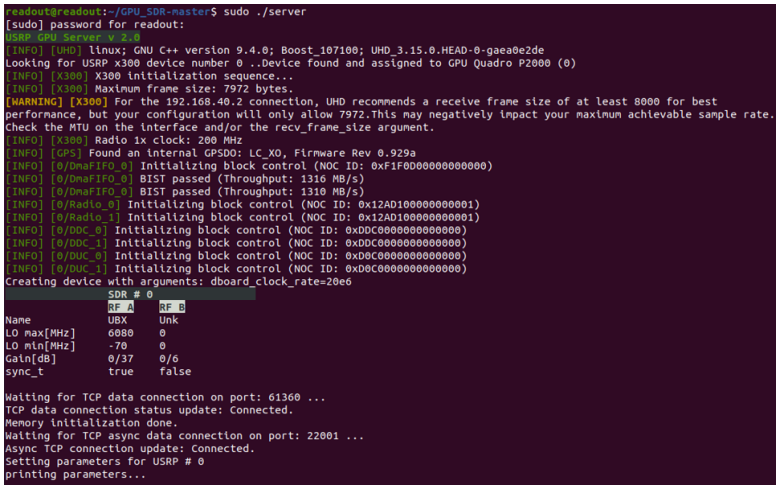

Figure 3.4: Starting GPU server

This session use the pyUSRP library, which includes all of the necessary functions for VNA (Vector Network Analyzer), noise acquisition, delay, and resonator measurements. The measurements are carried out using Python methods, whereas the server itself runs C++ code. To assure the use of Python version 2.7, which is now old and unsupported, our configuration required the use of a Python virtual environment. The process of finding the requisite versions for Python modules, the GCC compiler, the Boost library, and the UHD library proved to be a significant challenge throughout the project. This occurred because the method used was developed in 2019, and later versions have become obsolete and no longer available. Attempts to use automatic setup were unsuccessful because specific modules were missing from some versions. The setup was therefore manually configured. This necessitated testing and confirming the compatibility of multiple versions in order

to install the algorithm for Ubuntu's current setup and requirements, as well as the required libraries. This goal was accomplished by using a virtual environment. This was done to ensure that only manually installed versions are used and to avoid automatic system updates.
Python session is started using virtual environment and the fist step is to import the puUSRP library which allows to connect to the server and perform meaurements as shown below.

```
readout@readout:~/GPU_SDR-master$ source ~/venv/bin/activate
(venv) readout@readout:~/GPU_SDR-master$ python
Python 2.7.18 (default, Jul  1 2022, 12:27:04)
[GCC 9.4.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import pyUSRP as u
u>>> u.Connect()
Async command thread:
Socket binding [Errno 111] Connection refused, Retrying...
RX sync data thread:
Async command thread:
Socket binding [Errno 103] Software caused connection abort, Retrying...
RX data sync connected.
Async command thread:
Async data connected
True
>>> filename = u.measure_line_delay(100e6, 50e6, 'B')
Measuring line delay...
Saving on file: USRP_Delay_20240226_103556.h5
Using front-end: 'B'
Using sample rate: 100.00 MHz
Using LO frequency: 50.00
  0%|                                                                                |Ack message received from the server: Message received
100%|################################################################################|Async message from server: Measure finished
100%|################################################################################|
Cleaning data queue...
Queue cleaned of 0 packets.
H5 file closed succesfully.
Line delay acquisition terminated.
```

Figure 3.5: Connecting to the server and performing VNA measurements suing python session from virtual environment

Once the environment has been created, a Python script is used to control and interact with Universal Software Radio Peripheral (USRP) device.

```
source ~/venv/bin/activate
python
import pyUSRP as u
u.Connect()


filename = u.measure_line_delay(100e6, 50e6, 'B')
delay = u.analyze_line_delay(filename, True)
u.write_delay_to_file(filename, delay)
u.load_delay_from_file(filename)

vna_filename = u.Single_VNA(0, 100e6, 10, 5000, 20, Rate = 100e6, RF = 50e6,
  Front_end = 'B')
u.VNA_analysis(vna_filename)
u.vna_fit(vna_filename)
u.plot_VNA(vna_filename, backend = "plotly", plot_decim = None)
u.plot_resonators(vna_filename, reso_freq = None, backend = 'plotly')

rf_freq, tones = u.get_tones(vna_filename)
import numpy as np
tones = np.asarray(tones)

noise_filename = u.Get_noise(tones, 10, 100e6, decimation = 1, RF = 50e6, Front_end =
  'B', gain = 20)

u.copy_resonator_group(vna_filename, noise_filename)

u.diagnostic_VNA_noise(noise_filename, noise_points = None, VNA_file = None, ant =
  "B_RX2", backend = 'matplotlib')
```

Figure 3.6: Self written Python script used to perfrom measurements of the RF signal

The script's initial phrase establishes a relationship with the USRP, which prepares for a series of measures. The script-initiated function runs a systematic scan of a specified frequency range and stores the resulting delay data in order to measure signal delay along a specific pathway. Following that, the collected data is analyzed to better understand the features of signal latency. The evaluated results are carefully saved in files for easy retrieval and long-term preservation. The Vector Network Analyzer (VNA) examines the system under consideration by analyzing the influence of components such as amplifiers and filters on signals and providing useful information about their features. These measurements cover a wide range of frequencies and configurations to provide a full assessment of the system's efficacy. Modern plotting libraries are used to visually portray gathered data while also reviewing it for rapid comprehension in order to provide intricate details and increase clarity. Measurements of noise are as crucial as VNA measurements. The script isolates specific frequencies from VNA data and quantifies the related noise quantities while taking precision-ensuring variables into account, such as decimation rate and averaging. Further insights into the system's attributes can be obtained by comparing the noise data to the resonator data and identifying any correlations or patterns. Diagnostic procedures are performed to verify measurement precision and reliability.

Chapter 4

Results

4.1 Readout of GHz frequency signals

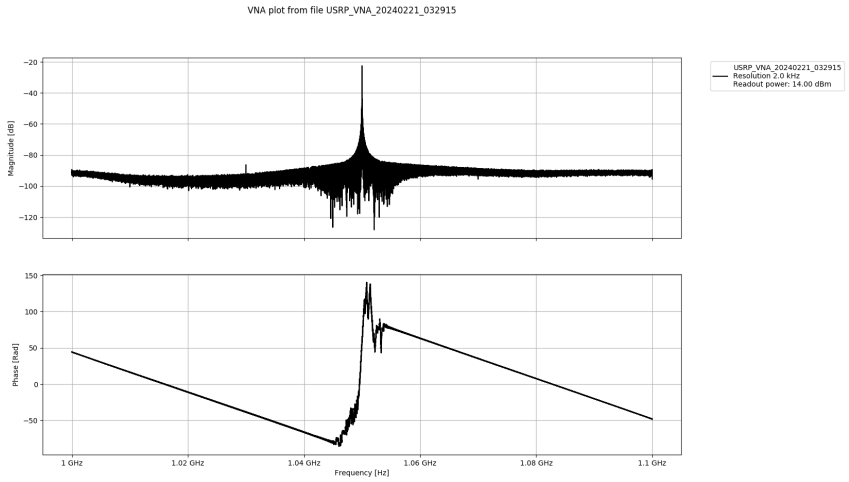

Figure 4.1: VNA analysis of 1GHz sinusoid signal using front end B

The presented plot is the result of a Vector Network Analyzer test with a USRP, used to readout the signal with a frequency of 1GHz. Typical frequencies for MKIDs generated signal are in the range 1-4 GHz, which emphaisize that the following test is predictor if the readout is ready to be employed in MKIDs experimental setup. The plot depicts the frequency spectrum using two graphs: the upper graph shows the amplitude in decibels (dB), while the lower graph shows the phase in radians (rad).

A notable surge can be seen on the magnitude map at around 1.025 GHz. The peak represents the sinusoidal signal under inquiry, and its small breadth indicates a significant quality factor, also known as the Q factor. This suggests that the signal

has negligible energy dissipation in comparison to its frequency, which is typical of a resonant system with low damping. The size of the signal far exceeds the expected noise level of -100 dB, indicating a clear and reliable reading.

The phase diagram shows a continuous reduction across the whole spectrum, which is indicative of a swept-frequency response. Resonance is defined as an abrupt discontinuity (phase shift) that occurs at the frequency where the magnitude peaks. It is assumed that a phase shift will occur as the signal approaches the frequency at which resonance occurs.

These results indicate that the method used to interpret the 1 GHz sinusoidal signal is precise. The existence of a prominent peak in the magnitude plot, along with a phase shift at the expected frequency, confirms that the signal is detected with outstanding accuracy. Furthermore, the peak's strong signal-to-noise ratio suggests excellent RF signal readout quality. However, it is crucial to identify the occasional appearance of anomalous peaks in the magnitude plot, since they could represent tiny reflections or interference inside the system. These oddities, however, appear to have no bearing on the fundamental signal of concern.

In addition, this readout was tested both front ends of the USRP and it was found that front end B performs better in terms of noise.

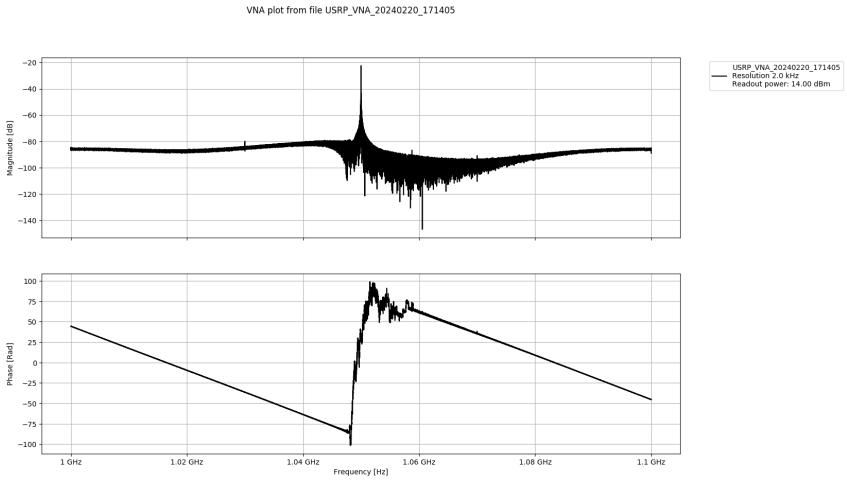

Figure 4.2: VNA analysis of 1GHz sinusoid signal using front end A

In conclusion, the VNA plot shows that a 1 GHz sinusoidal signal was read precisely and with good quality. The observed phase shift and strong peak in the figure indicate that the USRP-based RF signal measurement gear is functioning well.

4.2 Detection and measurements on 80 MHz signal

The VNA plot gives detailed information on the transmission and reflection characteristics of the radio frequency (RF) system over a wide frequency range. The plot shows a largely level baseline, with a notable peak at around 80 MHz. This peak's frequency is most likely the resonant frequency of one of the components in the MKID arrangement. It could be the resonator or a feature of the feedline component. To accurately interpret MKIDs' reactions to incoming photons, a full understanding of resonance frequencies is required. This is because these frequencies will be utilized to identify and analyze the MKID responses. A phase response plot is displayed beneath the VNA magnitude plot. This graph shows how the phase of the signal relates to its frequency. In MKID applications, phase response provides useful information on the dispersion properties of the transmission line or resonators employed. The presence of a phase shift at the resonant frequency allows for the identification between several resonant modes. This is something that may be expected.

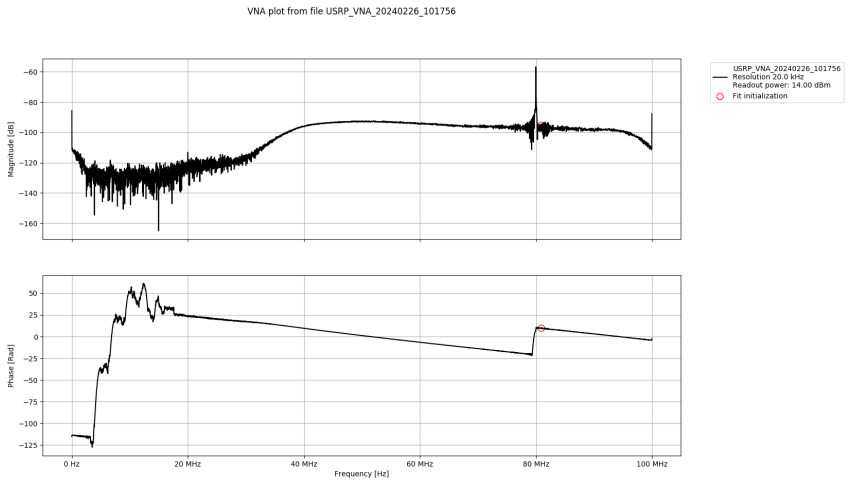

Figure 4.3: VNA analysis of 80GHz sinusoid signal

The resonator plot is a sophisticated I-Q plot that is critical to the operation of MKID. The graphic includes both in-phase and quadrature components. This program plots the resonator responses on the complex plane. Individual loops can sometimes represent the behavior of a single resonator. The interaction of photons with an MKID alters the inductance of the resonator, causing the loop to shift in the I-Q plane. To have a better knowledge of the quality factor (Q) and coupling efficacy of the resonator, the dimensions and positioning of these loops are taken into account. The graph on the right can be used to determine frequency shift (Δf) versus phase. This information is useful for modifying the system and determining the MKIDs' operational bandwidth.

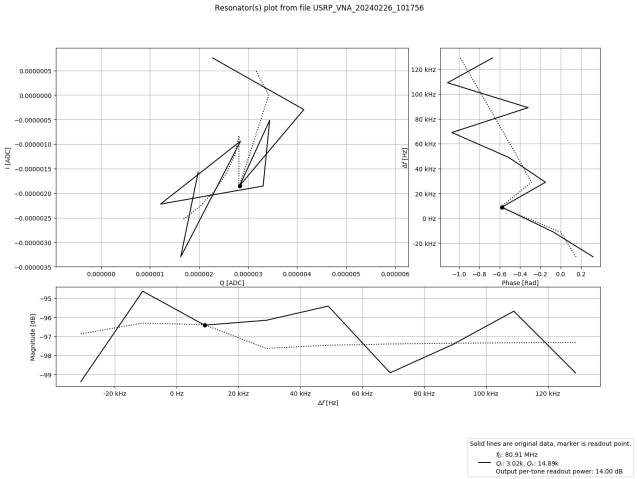

Figure 4.4: Resonators plot

The diagnostic map displays the system's noise characteristics overlaid on the averaged noise acquisition and VNA traces. High sensitivity and accuracy in photon detection with MKIDs require low noise levels. Using this diagram, researchers can determine whether the electronic noise level is low enough to not interfere with the signals emitted by MKIDs. An additional diagnostic plot provides for a complete evaluation of the noise and the system's response, which may indicate the level of stability the system maintains over time or under varied conditions. Stability is critical for MKIDs since deviations can impair photon detection accuracy.

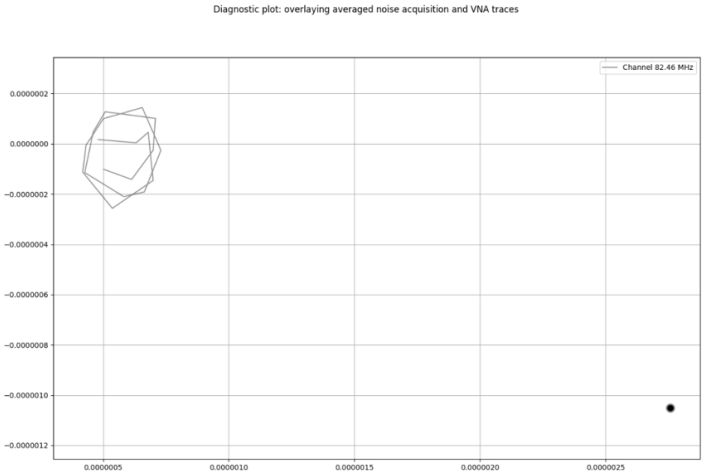

Figure 4.5: Noise aquisition

The script output for line delay analysis is used to account to any possible time delays in the signal path. Given its ability to influence the temporal preci-

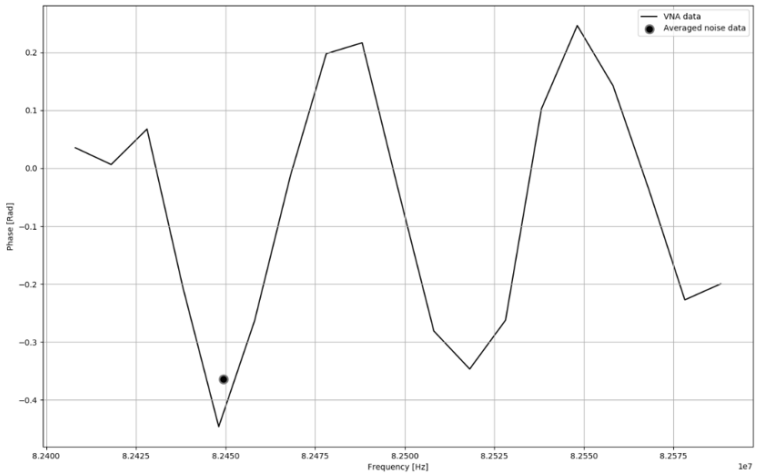

Figure 4.6: Plot of VNA data indication average noise

sion of photon detection, it is critical to thoroughly explore and reduce this delay in MKID models. By combining these plots and analyses, the testing approach

```
>>> delay = u.analyze_line_delay(filename)
Analyzing line delay info form file: 'USRP_Delay_20240226_103556' ...
Shape returned from openH5file(USRP_Delay_20240226_103556.h5) call: (1, 5000) is (channels,samples)
Coefficient is: 1.00000002e-10
Max low freq found: 10000.0
Delay found 1000 ns
>>> u.write_delay_to_file(filename, delay)
Storing a delay of 1000 ns for a rate of 100 Msps
```

Figure 4.7: Line delay measurements

demonstrated that the system exhibits the desired resonance characteristics, the noise environment is controllable, and the components, such as transmission lines, are well understood in terms of delay qualities. This demonstrates that the system was properly created and is operating within the expected parameters for MKID readout. Given this, the system is suitable for use in an authentic experimental configuration to detect photons with MKIDs, and there is a reasonable expectation that it will be reliable and accurate.

Chapter 5

Conclusion

This study has laid a solid framework for the potential integration of GPU-accelerated readout systems into microwave kinetic inductance detectors. We have begun to address the inherent limitations of FPGA-based systems, such as their inflexibility and high power consumption, by investigating enhanced GPU capabilities. Our research proposes a fresh and effective alternative for MKID applications, which, while not yet confirmed in real experimental settings, shows potential theoretical benefits. The initial results of adding GPU acceleration in MKID readout systems demonstrate a possible ability to handle large arrays of detectors and perform sophisticated computations with increased speed and efficiency. This result suggests a fundamental shift in how MKID technology may be used in the future, potentially increasing the detection and analysis of electromagnetic signals in a variety of scientific domains. The subsequent investigation will involve physically incorporating this reading method, which uses GPU acceleration, into practical MKID settings in order to objectively assess its effectiveness. This will allow for a direct comparison with conventional FPGA-based systems, potentially verifying the theoretical advantages of GPU acceleration. Additional research and innovation may lead to the development of more advanced and responsive detectors, improving our understanding of astrophysics, quantum computing, and materials science. To summarize, this study shows that GPU-accelerated MKID readout systems are both feasible and theoretically advantageous. Future experimental tests will demonstrate the actual impact and performance gains. GPU technologies are continually advancing, improving the performance and capacity of MKID systems. This propels them to the forefront of scientific research and innovation.

Bibliography

[1] Joris van Rantwijk et al. "Multiplexed Readout for 1000-Pixel Arrays of Microwave Kinetic Inductance Detectors". In: *IEEE Transactions on Microwave Theory and Techniques* 64.6 (2016), pp. 1876–1883. DOI: 10.1109/TMTT.2016.2544303.

[2] Omid Noroozian et al. "Crosstalk Reduction for Superconducting Microwave Resonator Arrays". In: *IEEE Transactions on Microwave Theory and Techniques* 60.5 (2012), pp. 1235–1243. DOI: 10.1109/TMTT.2012.2187538.

[3] J. Zmuidzinas and P.L. Richards. "Superconducting detectors and mixers for millimeter and submillimeter astrophysics". In: *Proceedings of the IEEE* 92.10 (2004), pp. 1597–1616. DOI: 10.1109/JPROC.2004.833670.

[4] Lorenza Ferrari et al. "Antenna Coupled MKID Performance Verification at 850 GHz for Large Format Astrophysics Arrays". In: *IEEE Transactions on Terahertz Science and Technology* 8.1 (2018), pp. 127–139. DOI: 10.1109/TTHZ.2017.2764378.

[5] *ALMA*. Accessed September 27, 2023. 2023. URL: https://www.almaobservatory.org/en/about-alma/.

[6] Masato Naruse et al. "Optical Efficiencies of Lens-Antenna Coupled Kinetic Inductance Detectors at 220 GHz". In: *IEEE Transactions on Terahertz Science and Technology* 3.2 (2013), pp. 180–186. DOI: 10.1109/TTHZ.2012.2237029.

[7] Mohsen Hosseini, Wei-Ting Wong, and Joseph C. Bardin. "A 0.4–1.2 GHz SiGe Cryogenic LNA for Readout of MKID Arrays". In: *2019 IEEE MTT-S International Microwave Symposium (IMS)*. 2019, pp. 164–167. DOI: 10.1109/MWSYM.2019.8701100.

[8] Matthias Arndt et al. "A Data Acquisition System for Kinetic-Inductance Detectors". In: *IEEE Transactions on Applied Superconductivity* 26.3 (2016), pp. 1–4. DOI: 10.1109/TASC.2016.2540241.

[9] K. J. Ramos, L. H. Arnaldi, and L. Tosi. "Towards a Low-Cost Readout System for Arrays of Cryogenic Detectors". In: *2023 Argentine Conference on Electronics (CAE)*. 2023, pp. 19–23. DOI: 10.1109/CAE56623.2023.10086976.

[10] Lorenzo Minutolo et al. “A Flexible GPU-Accelerated Radio-frequency Readout for Superconducting Detectors”. In: *IEEE Transactions on Applied Superconductivity* 29.5 (2019), pp. 1–5. DOI: 10.1109/TASC.2019.2912027.

[11] Ran Duan et al. “An open-source readout for MKIDs”. In: *Proceedings of SPIE - The International Society for Optical Engineering* 7741 (July 2010). DOI: 10.1117/12.856832.

[12] Beatriz Aja et al. “Analysis and Performance of Lumped-Element Kinetic Inductance Detectors for W-Band”. In: *IEEE Transactions on Microwave Theory and Techniques* 69.1 (2021), pp. 578–589. DOI: 10.1109/TMTT.2020.3038777.

[13] H. McCarrick et al. “Horn-coupled, commercially-fabricated aluminum lumped-element kinetic inductance detectors for millimeter wavelengths”. In: *Review of Scientific Instruments* 85.123117 (2014). DOI: 10.1063/1.4903855.

[14] Jinxuan Chen et al. “A Novel GPU Acceleration Algorithm Based on CUDA and MPI for Ray Tracing Wireless Channel Modeling”. In: *2023 IEEE Wireless Communications and Networking Conference (WCNC)*. 2023, pp. 1–6. DOI: 10.1109/WCNC55385.2023.10118847.

[15] Sean McHugh et al. “A readout for large arrays of microwave kinetic inductance detectors”. In: *The Review of scientific instruments* 83 (Apr. 2012), p. 044702. DOI: 10.1063/1.3700812.

[16] K. J. Ramos, L. H. Arnaldi, and L. Tosi. “Towards a Low-Cost Readout System for Arrays of Cryogenic Detectors”. In: *2023 Argentine Conference on Electronics (CAE)*. 2023, pp. 19–23. DOI: 10.1109/CAE56623.2023.10086976.

[17] Samuel W. Belling et al. “A Frequency Domain Multiplexing Technique for Multi-Channel Detector Instrumentation”. In: *2018 IEEE Nuclear Science Symposium and Medical Imaging Conference Proceedings (NSS/MIC)*. 2018, pp. 1–6. DOI: 10.1109/NSSMIC.2018.8824306.

[18] Eriko Nurvitadhi et al. “Accelerating Binarized Neural Networks: Comparison of FPGA, CPU, GPU, and ASIC”. In: *2016 International Conference on Field-Programmable Technology (FPT)*. 2016, pp. 77–84. DOI: 10.1109/FPT.2016.7929192.

[19] Svetlin A. Manavski. “CUDA Compatible GPU as an Efficient Hardware Accelerator for AES Cryptography”. In: *2007 IEEE International Conference on Signal Processing and Communications*. 2007, pp. 65–68. DOI: 10.1109/ICSPC.2007.4728256.

[20] John Nickolls and William J. Dally. “The GPU Computing Era”. In: *IEEE Micro* 30.2 (2010), pp. 56–69. DOI: 10.1109/MM.2010.41.

[21] Lingze Zhang, Yongxing Du, and Daocheng Wu. "GPU-Accelerated FDTD simulation of human tissue using C++ AMP". In: *2015 31st International Review of Progress in Applied Computational Electromagnetics (ACES)*. 2015, pp. 1–2.

[22] Shuai Che et al. "Accelerating Compute-Intensive Applications with GPUs and FPGAs". In: *2008 Symposium on Application Specific Processors*. 2008, pp. 101–107. DOI: 10.1109/SASP.2008.4570793.

[23] Shuo Chen and Xiaoming Li. "A hybrid GPU/CPU FFT library for large FFT problems". In: *2013 IEEE 32nd International Performance Computing and Communications Conference (IPCCC)*. 2013, pp. 1–10. DOI: 10.1109/PCCC.2013.6742796.

[24] Zhicheng Zhao and Yaqun Zhao. "The Optimization of FFT Algorithm Based with Parallel Computing on GPU". In: *2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*. 2018, pp. 2003–2007. DOI: 10.1109/IAEAC.2018.8577843.

[25] Chris Harris, Karen Haines, and Lister Staveley-Smith. "GPU accelerated radio astronomy signal convolution. Experimental Astronomy, 22(12), 129-141". In: *Experimental Astronomy* 22 (Oct. 2008), pp. 129–141. DOI: 10.1007/s10686-008-9114-9.

[26] Ettus Research. *USRP X310 High Performance Software Defined Radio*. Product Brochure. Ettus Research. URL: https://www.ettus.com/all-products/x310-kit/ (visited on 04/20/2024).

[27] Ettus Research. *UBX 10 MHz - 6 GHz Rx/Tx, 160 MHz BW*. URL: https://www.ettus.com/all-products/ubx160/ (visited on 04/20/2024).