EE4110: B-Tech Project Phase 1

INDIAN INSTITUTE OF TECHNOLOGY PALAKKAD

Final Report

Project Topic:

Development of a High Frame Rate Portable Ultrasound System

Arun Kumar V **122001049**

Advisor: **Dr. Mahesh R Panicker**
Associate Professor
Electrical Engineering

Acknowledgements

I want to express my heartfelt thanks to Dr. Mahesh R Panicker, my project mentor, for his incredible guidance and constant support throughout the project. I'd also like to extend my sincere appreciation to Mr. Madhavanunni for his technical discussions, guidance, and assistance during the experiments and imaging process.

I'm grateful to the Physical Chemistry and Biology lab, Centre for Computational Imaging lab, CSquare Innovation lab, and the research groups associated with them for providing access to their state-of-the-art facilities and resources.

I also want to express my gratitude to Texas Instruments and E-Lagori for supplying the necessary PCB boards and for their assistance.

Contents

1 Introduction

Ultrasound imaging is a well-established clinical imaging technique providing real-time, quantitative anatomical and physiological information in humans. The lack of ionizing radiation and relatively low purchase and maintenance costs result in it being one of the most affordable clinical imaging techniques with increasing use for guiding interventional clinical procedures.

However, despite its widespread use and numerous advantages, conventional ultrasound imaging does come with some limitations. Traditional ultrasound machines tend to be bulky and require a dedicated imaging suite, hindering their accessibility in certain clinical settings like war fields, ambulatory scenarios, etc. A Portable Ultrasound Imaging System which can also be referred to as a Point of Care (POC) diagnostic ultrasound imaging system can be said to be a solution to the drawbacks of traditional ultrasound imaging systems.

The Point of Care Ultrasound Systems (POCUS) brings advanced imaging capabilities directly to the bedside [1], they reduce the time between imaging and diagnosis, leading to a faster treatment. A programmable POCUS system could be made such that it offers a variety of imaging presets and post-processing options which could lead to improvement in image quality and diagnostic information.

However, even with the widespread adoption of POCUS, there remain some areas that can be improved. For example, most commercial systems are limited to low frame rate (typically, less than 50 frames per second) imaging as it employs focused transmit. Tt can capture only one frame when imaging, restricting the amount of real-time data that can be acquired. The aim of this project is to transform the system into a higher frame-rate portable ultrasound system by updating the system's underlying codes and algorithms.

2 Literature Review

The system-on-chip designs over the past few years have dropped 80% in both power consumption and size, these advancements enable the integration of complex circuits and active electronic components into a smaller area leading to the development of smart ultrasound probes or Universal Serial Bus (USB) Probes, which can push ultrasound technology into POC applications[2].

With the development of open platforms and programmable solutions, the field of ultrasound imaging has had significant advancements. New platforms and technologies are making it easier and cheaper to develop new ultrasound imaging devices. These devices are smaller, more powerful, and more flexible than traditional ultrasound scanners. They can be used for POC applications and can be customized to meet the specific needs of different users. Towards this, open platforms for accelerating smart ultrasound transducer probe development were reported in [2, 3].

In recent years, there has been a growing interest in the development of handheld ultrasound devices, particularly for POC applications. These devices offer portability and convenience, making them suitable for various medical scenarios. Several Field Programmable Gate Array (FPGA) -based handheld ultrasound devices have been proposed[4, 5, 6, 7], each presenting unique features and advancements.

G.-D. Kim et. al. introduced a portable ultrasound system utilizing a single FPGA (Spartan 3, Xilinx Inc.) with a 16-channel configuration. Despite its compact design, this system achieved image quality comparable to a 32-channel ultrasound system through the extended aperture (EA) technique. The maximum frame rate of 30 fps demonstrates its capability for real-time imaging. However, limitations in size (245 $mm \times 190$ mm) and weight (560 g) hindered its application in handheld Point-of-Care Ultrasound (POCUS), and the dedicated equipment for display added to compatibility challenges[4].

Y. Lee et. al. proposed a tablet PC-based handheld POCUS using a single Xilinx Spartan FPGA. This system employed 16 channels, extendable to 32 channels through the EA method, achieving a maximum frame rate of 22 fps. The use of an 11.6-inch Tablet PC for display addressed some of the size and weight concerns, providing a more user-friendly handheld solution[5].

In another approach, Kim et. al. presented a smartphone-based POCUS with a 128-element transducer, a 32-channel scanner module, and a single Xilinx Artix 7 FPGA. The system utilized a commercially available smartphone for backend processing, leveraging the GPU for computational tasks. The interactive graphical user interface (GUI) on the smartphone displayed the ultrasound results, achieving a frame rate of 50 fps[6].

Similarly, S. Ahn et. al. proposed a smartphone-based POCUS with a low-cost Xilinx Spartan 6 FPGA. This 16-channel system, extendable to 32 channels using the EA method, achieved a maximum frame rate of 58 fps. The integration of the FPGA with a smartphone, coupled with a user-friendly interface, enhanced the device's accessibility and usability in clinical settings[7].

The ultrasound systems proposed here use Xilinx Artix-7 XC7A100T FPGA[8] with a footprint of $15mm \times 15$ mm and 100,000 logic cells which is sufficient for a 64-channel ultrasound smart probe. This FPGA can be reprogrammed to make digital circuits that fit the specific needs of smart ultrasound transducers and can be used for implementing beamforming and image processing algorithms.

Some other hardware includes the AFE5832LP (Swaroop Board)[9] by Texas Instruments, which is a highly integrated Analog Front End (AFE). The TX7332[10] is a high-performance transmitter for ultrasound with 32 input channels. Four of these transmitters are stacked to achieve a 128-input channel system. The AFE5832LP can sample up to 50 Mega Samples per Second (MSPS), and the TX7332 supports high-frequency ultrasound by achieving a frequency greater than 20MHz at -3dB bandwidth. The system is powered via a USB Type C (USB 3.1 Gen 1), which provides a data transfer speed of up to 5 Gb/s, and the USB power supply module supplies a voltage ranging from 1 to $\pm$ 80 V from a very low input voltage (5 V). All the devices in the system are designed to have fast power-up and power-down management to reduce overall power consumption.

The CDCE949, Phase Locked-Loop(PLL) programmable clock synthesizers provide synchronized clocks for the dc/dc converters at frequencies of 500 and 250 kHz. This synchronization helps in eliminating frequency modulation in the case of the presence of multiple clocks in the system. The CYUSB301X, USB controller handles data communication between the FPGA and a PC, Laptop, or Mobile devices.

Some applications of the smart ultrasound transducer probes are COVID-19 diagnosis which limited the practical use of traditional ultrasound systems usage by doctors who are fully covered by Personal Protective Equipment(PPE) [11], POCUS[1], Non-Invasive Neuro-Simulation[11] etc. These applications not only make diagnosis better but also open up new possibilities for treating patients and monitoring their health.

Despite the evident advantages and potential of these open platforms and programmable solutions, some challenges exist, including the need for tradeoffs based on factors such as the availability and cost of the high-voltage transmitter, MUX, AFE, power consumption, etc. The AFE5832LP and TX7332 from Texas Instruments achieve low power and high integration, which assists in making these tradeoffs. Additionally, the FPGA has been programmed in a way that reduces power consumption.

In conclusion, the papers collectively present a comprehensive overview of the evolving landscape in the development of smart ultrasound transducer probes. Open platforms and programmable solutions hold the promise of revolutionizing ultrasound imaging by enabling innovation, reducing costs, and expanding the accessibility of smart ultrasound transducers. However, ongoing efforts to address complexity and establish standards are essential to fully harness the potential of these platforms for advancing medical imaging and healthcare.

3 Work Done

The setup we have consists of the Swaroop board which is a reference design from Texas Instruments (TI) for a portable handheld ultrasound imaging system. It is based on the TI AFE5832LP[9] and TX7332[10] integrated circuits (ICs), providing a complete solution for the signal conditioning, beamforming, and imaging processing required for ultrasound imaging[9].

The Swaroop board is a compact and lightweight design that can be easily integrated into a wearable device. The system features a 64-channel low-noise receiver, 128 transmitter channels, an FPGA, and a USB Controller.

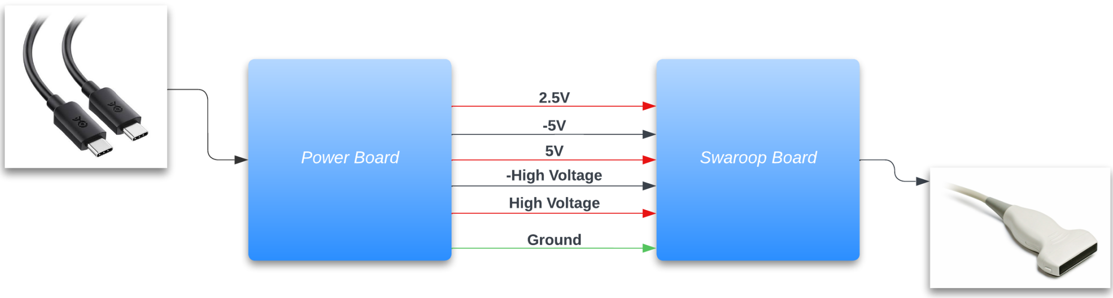
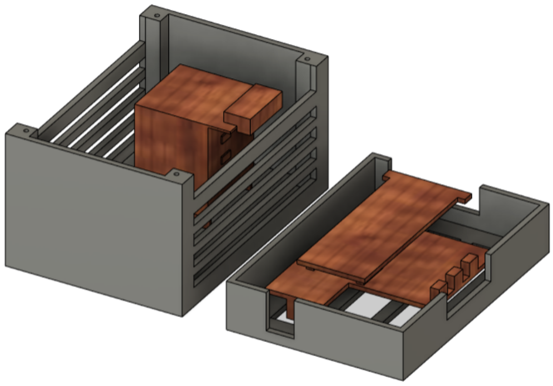


Figure 1: Block Diagram of the System

The Power Board is a High Voltage Generator*, which consists of four stacked standard power supply boards powered with a USB Type-C cable and capable of providing voltages up to $\pm$ 90 V. The voltage output of the High Voltage Generator can be controlled using Arduino IDE. The voltages supplied to the Swaroop board are 2.5 V, $\pm$ 5 V, and $\pm$ 90 V which are all generated by the High Voltage Generator. A linear ultrasound probe is connected to the Swaroop board, through which raw data is collected. This data is further processed using Matlab codes for additional diagnostics. The ultrasound probe used in this study is the ALS L12 - 5A Linear Array Probe with 128 elements and a center frequency of 7.5 MHz. The experimental setup employs plane wave transmission.

3.1 CAD Design

A concise and efficiently ventilated computer-aided design (CAD) was created utilizing Autodesk Fusion 360 and AutoCAD for the configuration of a printed circuit board (PCB). The assembly was constructed by laser cutting each side of the enclosure and then interlocking them together.



(a) Enclosure Design

(b) Enclosure

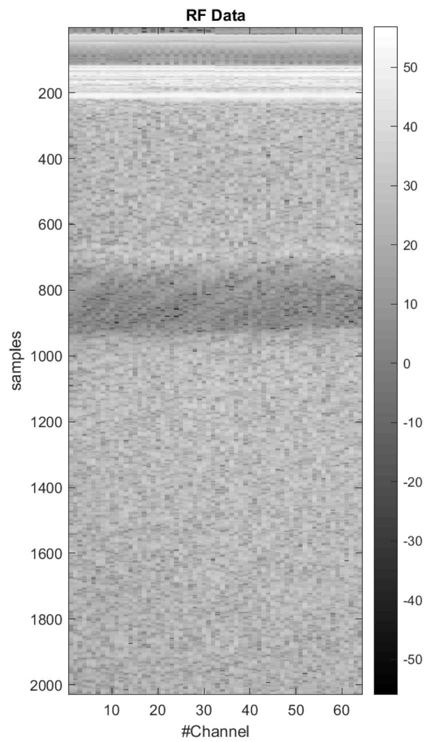
Figure 2: Enclosure design(Brown components are the system components for reference) and the Final product

3.2 Initial Experiments

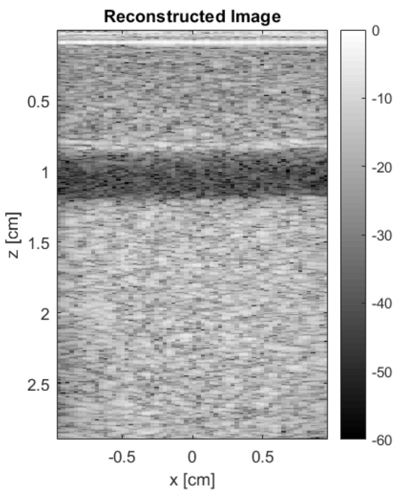
3.2.1 No Load Test of the Power Module

The High Voltage Generator was initially tested without connecting any loads to check whether it produced the desired voltages (2.5 $V \pm$ 5 V and $\pm$ 90 V are produced, The generator was powered using USB cables connected to a PC setup.

3.2.2 Load Test



(a) Raw Data

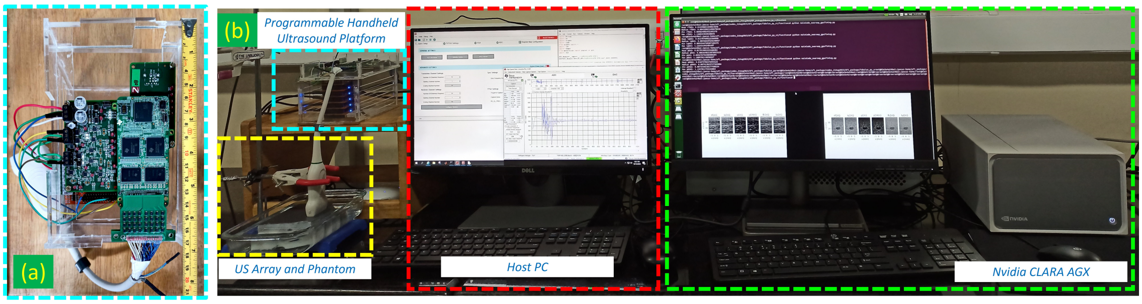


(b) Reconstructed Image

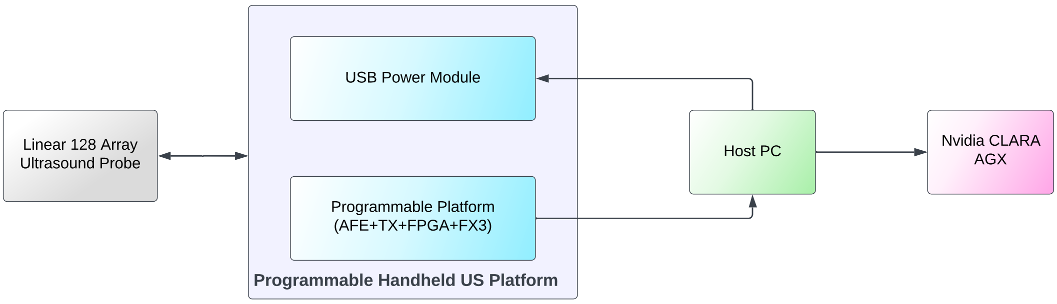
Figure 3: Captured raw data and the Reconstructed image of a single vessel phantom

The Swaroop Board was connected to the Power Board, and the voltages were checked, similar to the no-load test. Texas Instruments' high-speed data converter pro (HSDC Pro) software was used to analyze the analog performance of Analog-to-Digital Converters (ADCs) and Analog Front Ends (AFEs). The software was automated using National Instruments' LabVIEW and the Python 2.7 programming language, where users have the flexibility to set delays, imaging modes, aperture sizes, and other specifications. To test the proper functioning of the boards and the underlying algorithms, a sample test image was taken using the portable ultrasound setup. The image is of a tissue-mimicking phantom made during the last OELP. The CSV raw data(Figure 3a) saved by the Swaroop Board was reconstructed as an image using Matlab and is shown in Figure 3b.

3.3 Data Collection



(a) Imaging pipeline experimental setup[12]



(b) Proposed imaging pipeline

Figure 4: Captured raw data and the Reconstructed image of a single vessel phantom

An imaging pipeline was proposed as shown in Figure 6b, the raw dataset was captured using ALS L12 - 5A Linear Array Probe connected to the swaroop board with four 32-channel transmitters(TX7332[10]) and two highly integrated low-power 32-channel ultrasound Analog Front End (AFE, AFE5832LP [9]) to support 128-channel transmit and 64-channel receive. This data collected was transferred to a Host PC with HSDC Pro, a program designed to aid in the evaluation of TI high-speed data converter through a field-programmable gate array (FPGA) by the Cypress FX3[13] module. The FX3 provides high-speed radio frequency (RF) data transfer (throughput of up to 5 Gb/s). This received RF data was further pipelined into Nvidia CLARA AGX[14], commonly used for medical imaging systems for accelerated beamforming.

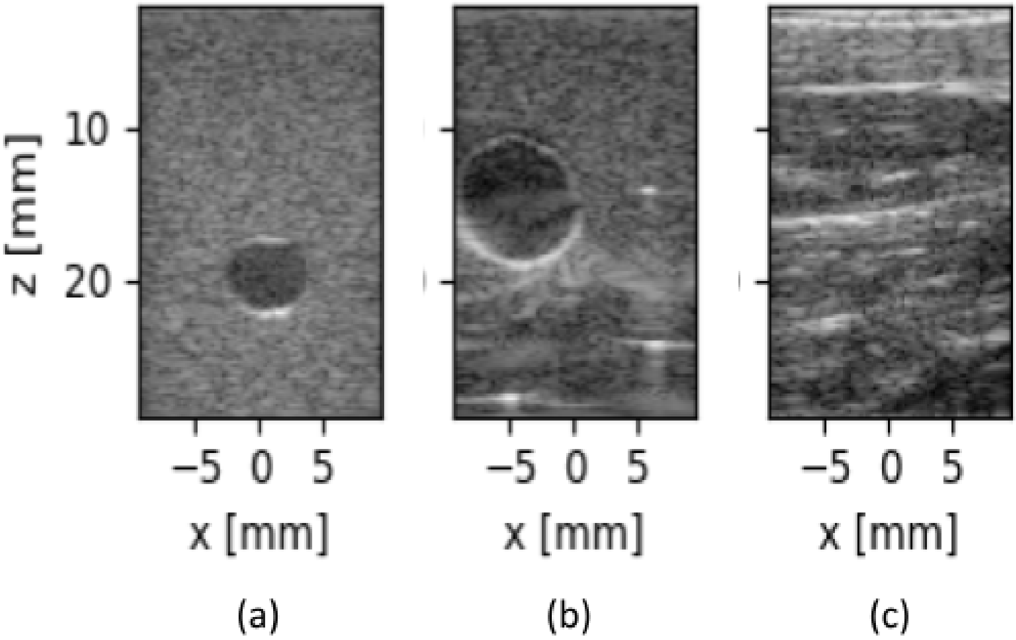
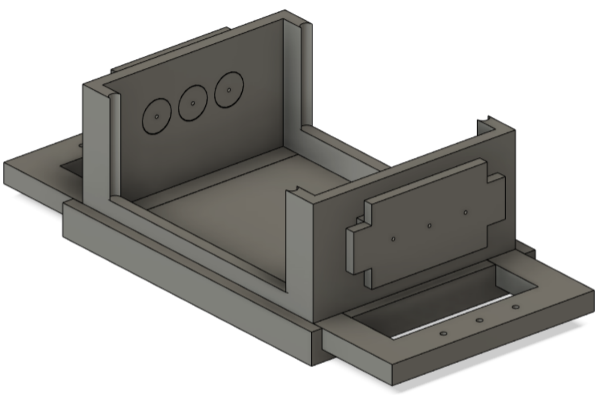


Figure 5: Experimental Results: (a)Reconstructed image of an *in-vitro* phantom having a vessel at 20mm depth.(b) Reconstructed image of an *in-vitro* phantom having anechoic cyst and point targets. (c) Reconstructed images from a sample *in-vivo* scan from brachioradialis of a healthy subject.

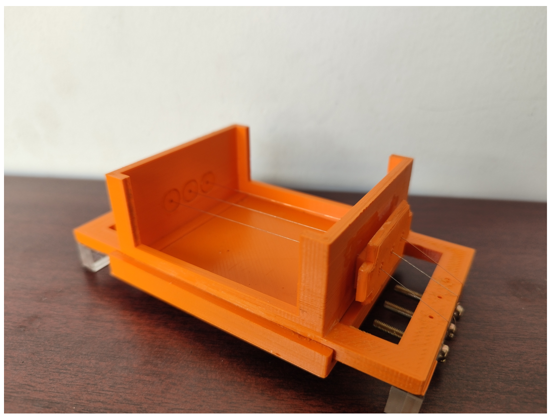
The Figure 5(a) depicts the reconstructed data from a phantom mimicking human tissue with a vessel at a 20mm depth from the surface. Figure 5(b) illustrates the reconstructed data from a phantom featuring an anechoic cyst at around 15mm depth from the surface and several point targets. The final figure, Figure 5(c), illustrates the reconstructed image from a scan of the brachioradialis muscle in a healthy individual. These data were collected and reconstructed using the proposed imaging pipeline.

3.4 Phantom Creation

For testing of the imaging pipeline, a microvascular phantom creation was initiated.



(a) Phantom mold CAD design



(b) Phantom mold

Figure 6: Captured raw data and the Reconstructed image of a single vessel phantom

The Figure 6 shows the CAD design and finished mold for a microvascular phantom with a vessel depth of 18mm, located 12mm from the top and bottom. The wire diameters used are 110μm, 140μm, and 230μm, mimicking real microvascular vessels present in the human body.

3.5 Continuous Data Capture

The existing system faces a limitation as continuous data capture is not possible. To address this, a MATLAB script was developed, which generates and writes datasets in real-time, mimicking ultrasound probe outputs and a Python script was developed to enable real-time processing of the raw datasets generated. This streamlined system ensures an uninterrupted flow of real-time data, overcoming previous limitations and enhancing adaptability in data processing. The following sections provide a detailed breakdown of the MATLAB and Python scripts involved.

Algorithm 1 Generate Random Binary Files(MATLAB)

Require: $filepath$ is the directory for storing binary files
Require: $filepath1$ is the updated directory for storing binary files
Ensure: Random binary files generated and stored

```
N ← []
A1 ← zeros(4060, 1)
j ← 0
stop_command ← false
while not stop_command do
    A1(1 : 2 : 4059, 1) ← randi([0, 255])
    A1(2 : 2 : 4060, 1) ← randi([0, 3])
    for i ← 0 to 63 do
        string1 ← sprintf('ADC Temp%d.bin', i)
        path ← append(filepath, string1)
        fileID1 ← fopen(path, 'w')
        fwrite(fileID1, A1)
    end for
    j ← j + 1
    disp(j)
    pause(0.5)
end while
```

This MATLAB script (Algorithm 1) generates an endless stream of binary files, each containing 64 sets of 4060×1 data points. It initializes an empty array N and a column vector A1 of zeros with dimensions 4060×1. The variable j is a counter, while stop_command is a boolean variable set to false. Within the primary loop, the script writes the odd indices of A1 with random values in the range [0, 255] and the even indices with random values in the range [0, 3]. Subsequently, it enters a parallel loop (parfor) to iterate over indices 0 through 63. For each iteration, it generates the filename following the pattern 'ADC Temp0.bin', 'ADC Temp1.bin', and so on. The binary file is then opened in write mode, and the data from A1 is written to the file using the fwrite function. This loop continues indefinitely, with the counter j incrementing, and a brief pause of 0.5 seconds between iterations. The script facilitates the continuous generation of evolving datasets.

Algorithm 2 Binary File Processing (Python)

Require: $filepath$ is the directory containing binary files
Require: $frame$ is the number of frames required
Ensure: Processed data in arrays N, M, and $Frame$

```
stop_flag ← False
i ← 0
N ← zeros(2030, 0, dtype = int16)
M ← zeros(2030, 0, dtype = int16)
Frame ← zeros(2030, 64, frame, dtype = int16)
m ← 0
procedure ProcessFiles
    file_mod_times ← {}
    while True do
        file_list ← sorted(os.listdir(filepath), key=natural_sort_key)
        for filename in file_list do
            f ← os.path.join(filepath, filename)
            current_mod_time ← os.path.getmtime(f)
            if file_mod_times[filename] ≠ current_mod_time then
                Read and process the file:
                    (Code for file processing)
                Update modification time:
                    file_mod_times[filename] ← current_mod_time
            end if
        end for
        Pause for 1 second:
            time.sleep(1)
    end while
end procedure
```

This Python script (Algorithm 2) is designed to continuously monitor a specified directory for changes in binary files and process the updated data in real-time. The script employs multithreading to ensure concurrent execution, allowing for continuous data processing as new files are generated. Upon detecting a modification in the directory, indicative of new binary files, the script reads and processes the binary data. The processing involves decoding the binary information considering the even indices values of the MATLAB generated array and storing the processed data in an array (N) . The script then organizes this data into a 3D array (Frame), where the third dimension represents frames defined by the user. When the specified frame limit is reached, the frame index resets, allowing for continuous overwriting from the beginning. The multithreaded architecture ensures that the script operates in real-time, adapting to evolving datasets. Additionally, the code tracks file modification times to optimize the processing of only the updated files. This integration of multithreading and file monitoring enables options for continuous data updates and real-time processing.

4 Conclusion

The development of a high frame rate portable ultrasound system using FPGA technology enables a significant advancement in the field of medical imaging. This innovative system offers numerous advantages, including real-time imaging capabilities, instant diagnosis, portability, etc. all of which can greatly benefit healthcare providers and patients alike.

The usage of FPGAs greatly improves signal processing and eases the tradeoff required between power consumption, cost, etc [11]. This opens up new possibilities for applications such as rapid diagnostic assessments, real-time guidance, and improved patient care.

The portability of the system enables medical professionals to remote areas, point-of-care (POC) situations, diverse clinical settings, in-field diagnosis, thus helping patients in regions with limited resources.

5 Future Work

The future work plan is to explore/exploit the possibilities of the system to continuously acquire the raw RF dataset and reconstruct the image in real-time or near-real-time. It would involve the following:

- Development of automation codes in Python
- A possible code/configuration changes in the FPGA to support the required continuous throughput of the RF data.
- Implement a feasible beamformer (Python-based) in CPU or GPU to realize a real-time or near-real-time image reconstruction.

References

[1] L. Lee and J. M. DeCara, "Point-of-care ultrasound," *Current Cardiology Reports*, vol. 22, pp. 1–10, 2020.

[2] X. Xu, S. A. Wala, A. Vishwa, J. Shen, K. Dijeesh, S. Devi, A. Chandak, S. Dixit, E. Granata, S. Pithadia, *et al.*, "A programmable platform for accelerating the development of smart ultrasound transducer probe," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 68, no. 4, pp. 1296–1304, 2020.

[3] X. Xu, S. A. Wala, A. Vishwa, J. Shen, K. Dijeesh, S. Devi, A. Chandak, S. Dixit, E. Granata, V. Nimran, *et al.*, "Open platform for accelerating smart ultrasound transducer probe development," in *2020 IEEE International Ultrasonics Symposium (IUS)*, pp. 1–4, IEEE, 2020.

[4] G.-D. Kim, C. Yoon, S.-B. Kye, Y. Lee, J. Kang, Y. Yoo, and T.-K. Song, "A single fpga-based portable ultrasound imaging system for point-of-care applications," *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, vol. 59, no. 7, pp. 1386–1394, 2012.

[5] Y. Lee, J. Kang, S. Yeo, J. Lee, G.-D. Kim, Y. Yoo, and T.-K. Song, "A new smart probe system for a tablet pc-based point-of-care ultrasound imaging system: Feasibility study," in *2014 IEEE International Ultrasonics Symposium*, pp. 1611–1614, IEEE, 2014.

[6] J. H. Kim, S. Yeo, M. Kim, S. Kye, Y. Lee, and T.-k. Song, "A smart-phone based portable ultrasound imaging system for point-of-care applications," in *2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pp. 1–5, IEEE, 2017.

[7] S. Ahn, J. Kang, P. Kim, G. Lee, E. Jeong, W. Jung, M. Park, and T.-k. Song, "Smartphone-based portable ultrasound imaging system: Prototype implementation and evaluation," in *2015 IEEE International Ultrasonics Symposium (IUS)*, pp. 1–4, IEEE, 2015.

[8] Advanced Micro Devices, "AMD Xilinx Artix-7 XC7A100T FPGA." https://www.xilinx.com/products/silicon-devices/fpga/artix-7.html. Last accessed 13 October 2023.

[9] Texas Instruments, "TI AFE5832LPEVM." https://www.ti.com/tool/AFE5832LPEVM.

[10] Texas Instruments, "TX7332." https://www.ti.com/product/TX7332. Last accessed 13 October 2023.

[11] X. Qian, R. Wodnicki, H. Kang, J. Zhang, H. Tchelepi, and Q. Zhou, "Current ultrasound technologies and instrumentation in the assessment and monitoring of covid-19 positive patients," *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, vol. 67, no. 11, pp. 2230–2240, 2020.

[12] A. N. Madhavanunni, V. Arun Kumar, Mahesh Raveendranatha Panicker, "A Portable Ultrasound Imaging Pipeline Implementation with GPU Acceleration on Nvidia CLARA AGX." https://arxiv.org/abs/2311.00482.

[13] Infineon Technologies, "Cypress FX3 CYUSB301X USB Controller." https://www.infineon.com/dgdl/Infineon-CYUSB301X_CYUSB201X_EZ-USB_FX3_SUPERSPEED_USB_CONTROLLER-DataSheet-v21_00-EN.pdf?fileId=8ac78c8c7d0d8da4017d0eca1e7442aa. Last accessed 13 October 2023.

[14] Nvidia, "NVIDIA Clara AGX." https://www.nvidia.com/en-gb/clara/intelligent-medical-instruments/. Last accessed 13 October 2023.