Low-Power Ultrasound Imaging on Mixed FPGA/GPU Systems

Damian Cacko
Laboratory of Professional Electronics
Institute of Fundamental Technological Research
Polish Academy of Sciences
Warsaw, Poland

us4us sp. z o.o
Warsaw, Poland
dcacko@ippt.pan.pl

Mateusz Walczak
Laboratory of Professional Electronics
Institute of Fundamental Technological Research
Polish Academy of Sciences
Warsaw, Poland

us4us sp. z o.o
Warsaw, Poland

Marcin Lewandowski
Laboratory of Professional Electronics
Institute of Fundamental Technological Research
Polish Academy of Sciences
Warsaw, Poland

us4us sp. z o.o
Warsaw, Poland

Abstract—**Portable and hand-held ultrasound imagers have the potential to revolutionize Point-of-Care medical diagnostics. There is great need for low-cost, portable scanners with extended battery life. In this paper, we focus on hardware-software partitioning in heterogeneous systems where both field-programmable gate array (FPGA) and graphics processing unit (GPU) resources are available. We present the architecture of a prototype test scanner for the evaluation of various hardware-software partitioning strategies. The system is equipped with the Intel Arria 10 FPGA and the Nvidia Tegra X2 mobile GPU. FPGA-based beamformers: Delay-and-Sum and Filtered Multiply-and-Sum, were implemented. These 32-channel beamformer blocks are integrated into a complete dataflow along with the data acquisition, RF filter, quadrature demodulator, and envelope detector. The designed dataflow allows one to allocate processing functions to either hardware (FPGA) or software (GPU) to explore various imaging scenarios and optimize power consumption. A dedicated measurement setup facilitates measuring power consumption of both FPGA and GPU. The developed setup will provide a reliable experimental system power characterization.**

Keywords—ultrasound imaging, ultrasound scanner, point-of-care ultrasound; beamforming, low-power, FPGA, GPU processing

I. Introduction

Due to its noninvasiveness and safety, the ultrasound imaging technique is the modality of choice for general medical diagnosis. Along with continuous and rapid development in microelectronics, the computing performance of modern ultrasound imaging systems continues to increase, making the implementation of complex and sophisticated imaging algorithms feasible. For the same reason, ultrasound systems become more and more integrated, which allows to scale them to portable sizes. In recent years, there has been a growing interest in point-of-care ultrasound, or using ultrasonography at a patient's bedside for diagnostic and therapeutic purposes [1]. It is used by various specialties: anesthesia, cardiology, critical care medicine, emergency medicine, pediatrics and many others [2].

Point-of-care ultrasound is efficient only if it provides high image quality in real-time while maintaining the compact size of the imaging device. This creates a challenge for engineers to scale the systems efficiently and to profile the ultrasound signal processing in terms of low-power consumption. Portable systems are usually battery-powered, and the success of a product is dictated by its battery life, so reducing power consumption is critical.

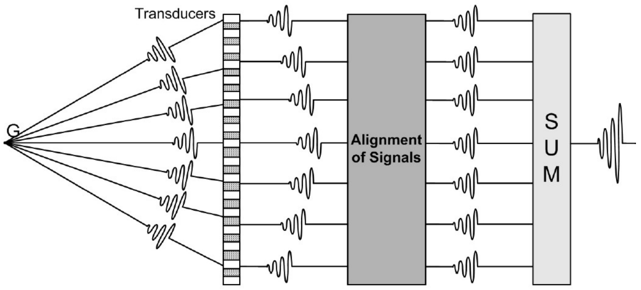

Fig. 1. Beamforming principle in phased-array ultrasound systems [4].

Power supplies for medical systems also have strong requirements for safety and quality that must be met. There have been extensive efforts to design low-cost portable ultrasound systems by changing the transducer design, the transmission and reception circuitry hardware implementation, or the beamforming algorithms [3]. A careful design of the transmit and receive circuits and the echo signal processing implementation approach are key aspects to achieving low-power operation while maintaining high frame-rate, high-quality imaging.

Typically, in modern ultrasound imaging systems, both portable and stationary, a phased-array probe is used to form the B-mode image. In these systems, to compensate for the low signal-to-noise ratio (SNR) of the signals received by the individual transducers within a probe, large transducer arrays ranging from 32-512 elements are used. The process called beamforming is applied to all received signals to provide spatial selectivity and construct high-resolution, high-contrast images. Fig. 1 depicts how signals are aligned to detect a focal point in a phased-array system. If the focal point G is under detection, the transmit signals reflect and arrive at the individual transducers at different times. The role of the beamformer is to appropriately delay the RF signals to align and sum them together. Frequently, before summing, an apodization is performed, realized by multiplying signals in each channel by a weighting function. In systems featuring dynamic focusing, during the receive phase, the beamformer changes the focal point continuously with the depth by adjusting the delays on all channels. These delays must be calculated for each position of the focal point. Delay values and apodization coefficients can be computed in real-time during the receive process [5] or can be pre-calculated before data acquisition and stored

978-1-5386-7114-6/18/$31.00 ©2018 IEEE

in memory [6]. Further essential signal processing performed in conventional ultrasound systems after summation involves filtering, detection, and log compression. The summed signal is band-pass filtered to attenuate the noise outside the band of interest. Since ultrasound imaging is done on an envelope of the signal, it should be extracted from the RF signal. This is called detection and is commonly performed in two stages. The first stage is to get a complex I/Q signal by using the Hilbert transform or quadrature demodulator, followed by low-pass filters on both the I and Q component. The envelope signal can be obtained by calculating the square root of the sum of the in-phase and quadrature signals' squares.

The B-mode ultrasound image is composed of multiple scanlines. The algorithm, as described above, is used to create a single scanline of the image and must be repeated for each scanline within the image. Image display is preceded by the scan conversion operation. The basic problem of scan conversion is to interpolate raw data to displayed data. In phased-array ultrasound systems, the envelope data is in polar coordinates. A coordinate transformation needed to interpolate the data accurately on the display grid depends on the display resolution.

One of the key challenges of ultrasound imaging systems design is real-time processing. For instance, a 256-channel system with 12-bit A/D converters sampling at 65 MSPS produces 23.25 GB/s of data throughput, which has to be processed in real-time. This results in a very high computation performance requirement of the digital processing unit. This requirement is challenging even for the most classic image reconstruction approach of delay-and-sum (DAS) beamforming (as described in this section), which is commonly used in portable point-of-care ultrasound devices. However, to improve image quality, many advanced imaging techniques that require even more computational performance have been proposed, such as: synthetic aperture methods [7], adaptive beamforming methods [8] or coded transmission [9].

II. Project Motivation

Over the last dozen years, researchers and ultrasound system designers have used field-programmable gate arrays (FPGAs) to implement high-speed digital signal processing, by utilizing their parallel processing capabilities and design flexibility. Modern FPGAs offer sufficient and comprehensive resources to implement complete multi-channel signal processing systems that operate with a sampling frequency from tens to over 100 MHz. Numerous scientific reports have been published about successful FPGA-based ultrasound imaging system development [4-6][10-14]. These systems differ in size, complexity, functionality and performance.

A new trend, observed in recent years, is the extensive use of graphics processing units (GPUs) to deliver high-performance computing into many applications [15]. The GPUs provide massive parallel execution resources and a high memory bandwidth. Language APIs such as Nvidia CUDA C and development environments made it possible to programme GPUs in a straightforward way without the need to map traditional OpenGL and DirectX APIs to implement general purpose operations. In [16], Che et al. presented a performance study of three diverse applications using FPGA, GPU and multicore CPU. Their results show that both FPGAs and GPUs can boost system performance by an order of magnitude or more, in comparison to CPU implementation.

The introduction of GPU processing makes it possible to perform all image reconstruction entirely in the software. Several research groups have reported successful ultrasound scanners with real-time imaging construction [17-19].

Lewandowski et al. [20] presented an initial development of an ultrasound hardware platform with GPU processing with an optimized GPU-to-CPU link for high frame-rate imaging. Choe et al. [21] reported development of a complete GPU-based real-time ultrasound system with software beamforming. Recently, GE Healthcare has introduced Vivid E95 cSound, a commercially available ultrasound scanner with software beamforming. In [22], Kulina et al. state that software beamforming in this scanner can provide higher quality imaging for endocardial border detection than high-end hardware-based beamformers.

Unfortunately, while providing remarkable performance improvements, the GPUs also exhibit high power consumption, which limits their usage mainly to stationary research platforms. However, recently, power-efficient GPUs such as Nvidia Tegra X2 (Nvidia Corporation, Santa Clara, USA) with a power consumption of only several watts have been introduced. The implementation of point-of-care ultrasound systems with GPU-based software processing can now be explored. The software approach is a convenient way to implement novel imaging algorithms into point-of-care devices. Therefore, our team has developed a system solution utilizing the Nvidia Tegra X2 for software-based imagers.

Although the main part of the processing is executed on the GPU in the software-based system, there is still a need for FPGAs supporting high-speed data capture and buffering [23]. The high-speed interface solutions currently available do not support a direct connection of multichannel ADCs to a GPU. As a result, the FPGA placed in the signal path is still capable of serving as both a piece of data buffering hardware and a computational accelerator at the same time. Since data capture and transfer uses usually only a small part of modern FPGA resources, some of the acceleration tasks can be shifted to the FPGA in order to:

- reduce the I/O bandwidth,
- free computing resources of the GPU, which can be used in further post-processing stages,
- reduce the total power consumption of the system.

Therefore, in power-efficient systems, the partitioning of signal processing between the FPGA and GPU should be explored. In this work, we propose a heterogeneous FPGA/GPU ultrasound test system with FPGA buffering which is optimized for direct raw RF data transfer to be processed within GPU. The system is convenient for the evaluation and optimization of power consumption depending on the applied hardware-software partitioning of the signal processing.

III. System Architecture

The high-level architecture of the heterogeneous FPGA/GPU ultrasound signal processing test system is shown in Fig. 2. The system is based on a real-time PCIe streaming architecture.

A. Hardware Platform

The system hardware (Fig. 3) integrates 128 transmit and 32 receive channels. The transmit section uses STHV800 (STMicroelectronics) octal 3-level ultrasound pulsers, which are able to produce ±90 V output bipolar square waveforms with up to 2 A peak current. An FPGA-based transmit beamformer controls each of the 32 TX channels and provides up to 512 unique firings definitions. Each firing configuration includes: output signal central frequency, number of cycles within a pulse, aperture setting and transmission delay individual for each transmit channel. The timing resolution of the TX beamformer is 5 ns.

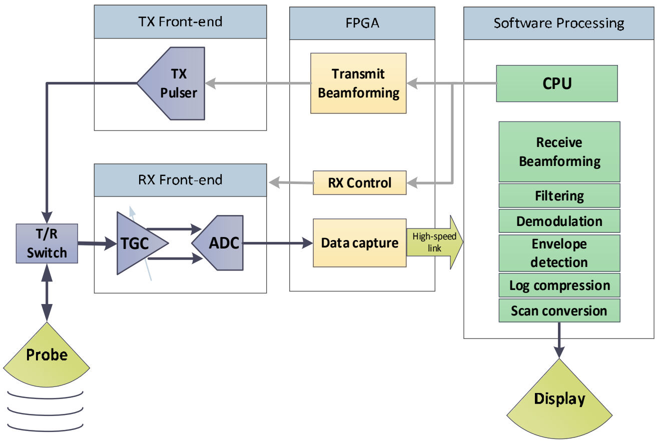

Fig. 2. Heterogenous FPGA/GPU ultrasound system architecture. In the diagram it is assumed that fully software-based beamforming is performed.

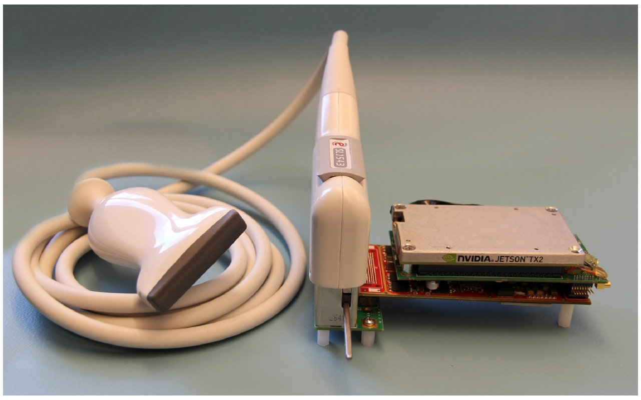

Fig. 3. A view of the hardware platform (us4us Ltd.) connected with the Esaote linear array probe. The main electronic modules are visible on the right-hand side: Jetson TX2 module with the Nvidia Tegra X2 GPU and the rest of the hardware – including FPGA, transmit and receive circuits – on the bottom.

The receiver hardware is based on two ADE58JD18 (Texas Instruments) chips, each consisting of 16-channels of ultrasound analog front-end (AFE). Each AFE channel includes a low-noise preamplifier, a time gain control (TGC) amplifier with a gain range of 40 dB, a 3rd order analog active low-pass filter with a configurable cut-off frequency and a 12/14-bit ADC running at 65/40 MSPS respectively. The AFE outputs data at 5 Gbps JESD204B interface with 2, 4, or 8 channels data per JESD lane. The gain of the TGC amplifier can be controlled during data acquisition to apply a custom time gain compensation profile. The AFE58JD18's internal registers are accessible using an SPI interface, and control AFE parameters such as: total gain, high-pass and low-pass cut-off frequencies, low-power/low-noise profiling settings, ADC resolution, output data format and many others.

The key component of both the transmit and receive section of the system is Intel Arria 10 FPGA, which:

- captures ADC output data in JESD204B interface from all channels and deserializes data for further processing or data transfer;
- forms the PCIe packets and transfers data out to the software processing;
- captures commands from the external software controller via PCIe link;
- configures the TX beamformer using SPI protocol between each transmission by updating settings to achieve a new wavefront according to transmission firing scenario;
- controls the TGC gain during receive phase,
- writes internal registers of AFE chips using the dedicated SPI interface;
- interfaces with external DDR4 memory to perform data buffering;
- generates synchronization signals for the rest of the system.

By using a high-speed PCIe link, raw data can be transferred in real-time to Nvidia Tegra X2 GPU, where it undergoes software processing. For demonstration purposes, an example PWI (Plane Wave Imaging) reconstruction algorithm was implemented. A complete digital signal processing flow includes: raw RF data filtering, demodulation, PWI reconstruction, envelope detection and log compression. All these steps produce a final B-mode ultrasound image at a rate of 50 fps.

B. FPGA-based DAS Beamformer Implementation

Because of its parallel nature and simple summing operations, the delay-and-sum (DAS) beamforming algorithm can be effectively implemented using FPGA. We have implemented a complete DAS beamformer in the FPGA. Its architecture is shown in Fig. 4.

Digitized and deserialized 14-bit @ 65 MSPS data from

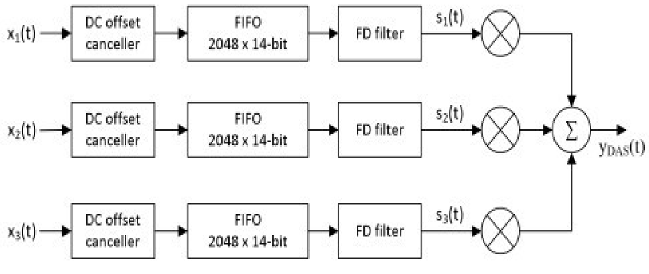

Fig. 4. The architecture of the DAS beamformer for FPGA. For the sake of brevity, only 3 of the total 32 channels are shown.

each AFE processing channel goes to the DC offset canceller block, which automatically detects the level of offset present in the signal and then subtracts this value from the signal. Two further modules are designed to apply a time-shift of the signals to re-align them. In this implementation, a combination of coarse and fine delay strategy is adopted. The coarse delay, expressed as an integer multiple of the clock period, is accomplished by using a FIFO buffer with maximum depth of 2048 samples. This allows to apply a maximum delay of around 31 μs. Actual FIFO depth used in an individual channel is controlled by a dedicated logic to achieve the desired coarse delay in this channel. A fine delay is generated using fractional delay (FD) filters, which are implemented as 8th order FIR filters with symmetric impulse response. An in-depth overview of FD filters and their characteristics can be found in [24]. In this design, the filter coefficients were derived by using a classic Lagrange interpolation formula:

$$h_d(n) = \prod_{\substack{k=0 \\ k \neq n}}^{N} \frac{\lfloor (N+1)/2 \rfloor + d - k}{n - k}, \quad n = 0,1,2,\dots,N$$

where $h_d\,(n)$ denotes the n^{th} tap coefficient of the FD FIR filter, N is the number of taps and d is the fractional part of the unit (clock) delay. To achieve a delay resolution setting of 1/10 of the clock period (~ 1.5 ns), nine different coefficient sets were calculated for different values of d (0.1, 0.2, ..., 0.9), which can be chosen to apply in the FPGA datapath according to fine delay desired in a specified channel. After delaying the signals, apodization is performed by weighting the signals in each channel. Weighting is implemented as multiplying signals by 16-bit weight factors. After that, all the signals are summed to form a single 18-bit RF signal.

To perform dynamic receive focusing, the developed beamformer is able to adjust the delays in all channels. A maximum of 128 delays (both coarse and fine) for each channel can be defined for each scanline to achieve multiple focal points. An internal counter controls the moments of changing delay settings, and its interval is configurable prior to the data acquisition.

If using the same assignments as in Fig. 4, the DAS beamformer output signal can be derived as:

$$y_{DAS}(t) = \sum_{i=1}^{N} a_i(t) \cdot x_i[t - \tau_{Di}(t) - \tau_{di}(t)],$$

where a_i is the apodization coefficient for i^{th} channel, while τ_{Di} and τ_{di} are coarse and fine delays respectively. In the case of dynamic focusing, each of these variables is generally a function of time.

Fig. 5 shows the rest of the digital signal processing implemented in the FPGA. The RF signal is pass-band filtered using a 96-tap finite impulse response filter (FIR) with a configurable coefficient set for maximum flexibility. As an example, by using this filter it was possible, during initial tests, to achieve the following response of the filter: pass-band response 1.5-4.5 MHz, pass-band ripple 0.01 dB, stopband attenuation 84 dB, and transition bands width 1.1 MHz. After filtering, the signal is demodulated using a quadrature I/Q demodulator. The shift frequency is controlled by setting the output frequency of the direct digital synthesis (DDS) sine/cosine wave generator.

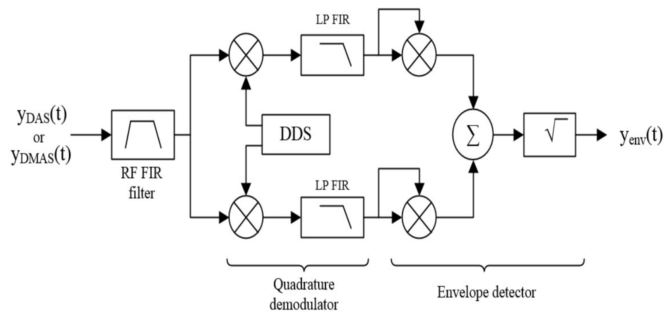

Fig. 5. The RF signal processing datapath for a FPGA implementation. DDS – Direct Digital Synthesis generator, LP FIR – low-pass finite impulse response filter).

The I and Q components are then low-pass filtered by the 96-tap FIR filters to remove the undesired products of demodulation at doubled carrier frequency. Moreover, during the filtering process, a decimation is performed by a factor of 2. An envelope signal is then derived from the filtered I and Q signals as a magnitude of a complex number.

C. FPGA-based FDMAS Beamformer Implementation

As an alternative to the classical DAS beamformer, we have also implemented a non-linear filtered delay-multiply-and-sum (FDMAS) algorithm. A dataflow of the improved DMAS algorithm proposed in [25] is shown in Fig. 6.

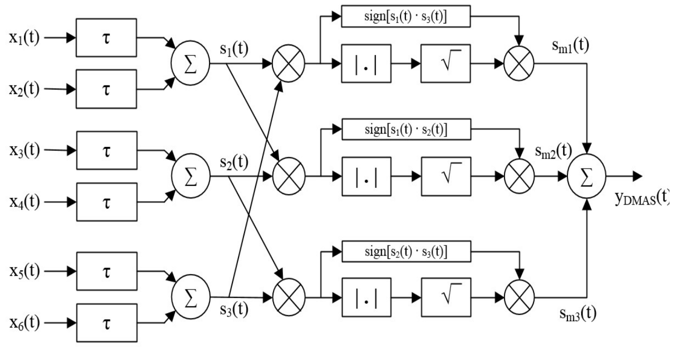

Fig. 6. The architecture of the FDMAS beamformer for FPGA. For the sake of brevity, only 6 out of 32 channels are shown. Blocks marked with τ are time delay blocks, which consist of an FIFO buffer and an FD filter as in the DAS beamformer.

In the first stage, signal processing involves time shifting all signals to re-align them (as in the classic DAS). The implementation of the delay block is the same as in the DAS beamformer. It consists of both an FIFO buffer and FD filter to provide coarse and fine delays respectively. It also features a dynamic focusing functionality. Once the signals are in phase, they are combinatorically coupled and multiplied. This operation requires a high number of multipliers to perform the operation in real-time. The number of multipliers K needed for N-channel FDMAS beamformer implementation is given by all possible pair combinations and can be derived as:

$$K = \binom{N}{2} = \frac{(N-1) \cdot N}{2}$$

To implement an 8-channel FDMAS beamformer, one would need 28 multipliers to design the multiplying stage. For 16, 32, 64 and 128-channel system, this number would be 120, 496, 2016, 8128. Due to the limited number of multipliers in the FPGA chip used (total of 312), in this study we present a 32-channel receive beamformer with a 16-channel multiplying stage. After time shifting, adjacent channels are summed and then reach the multiplying block, which consists of 120 parallel multipliers. The operation performed by the multiplying block can be mathematically interpreted as the auto-correlation function of the receive aperture [24]. In each of the 120 parallel datapaths, after multiplication, the square root of the absolute value is calculated and then the initial sign of the input value is applied to the result. The FDMAS beamformer output signal y_{DMAS}, using assignments as in Fig. 5., can be derived as:

$$y_{DMAS}(t) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \operatorname{sgn}\{s_i(t) \cdot s_j(t)\} \cdot \sqrt{|s_i(t) \cdot s_j(t)|}$$

Since the multiplication of signals of similar frequencies occurs in the datapath, the resulting spectra contain one component centered at $f = 0$ and another centered around twice the carrier frequency (harmonic). Therefore, the next step is a band-pass filter to cut-off the offset component and the high-frequency noise, while keeping the harmonic signal unaltered. Further, envelope detection is implemented in the same manner as presented in Fig. 5 for the DAS algorithm.

In both data processing approaches, it is possible to capture the digital signals at various stages of processing. The FPGA design supports capturing the envelope signal, the data after summation (bypassing demodulator and envelope detector), data after delay application, and original sampled data (bypassing the whole beamformer circuit).

Table 1 summarizes the resource utilization of a preliminary implementation of both the DAS and FDMAS beamformers in the Intel Arria 10 FPGA.

TABLE I. A RESOURCE UTILIZATION OF THE FPGA-BASED BEAMFORMERS IMPLEMENTATION.

Resource	**Utilization count / [%] of total available**	
	DAS	***FDMAS***
Adaptive logic modules (ALM)	13904 (22 %)	29940 (49 %)
Flip-flops	39128 (16 %)	60818 (24 %)
Hardware multipliers	260 (83 %)	312 (100 %)
Embedded RAM [Kb]	992 (11 %)	992 (11 %)

The utilization includes only the digital signal processing functionality, while other functional blocks implemented in the FPGA, such as JESD204B or PCIe interfacing (as listed in section III.A), consume in total: 18327 adaptive logic modules, 18502 flip-flops, 2978 Kb of embedded RAM. Both the DAS and FDMAS beamformers combined with this additional logic will fit in the chosen FPGA device. The next step of the design will be to integrate the described beamformers with these interfaces and software.

IV. CONLUSIONS AND FUTURE PERSPECTIVE

In this paper we have proposed a heterogeneous ultrasound system architecture, which brings GPU processing to portable devices for point-of-care applications. This flexible architecture allows for a comprehensive evaluation of various hardware-software partitioning strategies within such systems. The presented system allows to integrate a complete ultrasound signal processing chain and split the processing between the hardware (FPGA) and software (GPU). Although the system architecture was optimized for software-based beamforming, alternative FPGA-based DAS and FDMAS beamformers implementations were also introduced and described.

In the next step, we are planning to finalize the integration of the developed DAS and FDMAS beamformer blocks with the entire system, as well as evaluate various aspects of the hardware/software partitioning – such as frame rate, image quality, and power consumption. The corresponding beamforming algorithms will be implemented on the GPU to compare the hardware-based and software-based approaches.

Finally, an optimization of the hardware-software partitioning for various modes of operations will be performed. The presented hardware platform enables the testing and performance evaluation of novel beamforming algorithms in a mixed FPGA/GPU processing environment.

Power efficiency is one of the most important figures of merit for portable Point-of-Care imagers. The described approach gives an insight into the internals and trade-offs of hardware and software ultrasound signal processing. The platform may be used for both the evaluation and optimization of digital processing implementations, and as a target platform for a new class of portable scanners.

REFERENCES

[1] F. M. Abu-Zidan, "Point-of-care ultrasound in critically ill patients: Where do we stand?," J. Emerg. Trauma Shock, vol. 5, no. 1, pp. 70 - 71, 2012.

[2] C. Moore and J. Copel, "Point-of-care ultrasonography," New Engl. J. Med., vol. 364, pp. 749–757, 2011.

[3] J. M. Baran and J. G. Webster, "Design of Low-Cost Ultrasound Systems: Review," in Proc. IEEE Engineering in Medicine and Boilogy Society, pp. 792–795, 2009..

[4] M. Almekkawy, J. Xu, and M. Chirala, "An optimized ultrasound digital beamformer with dynamic focusing implemented on FPGA," in Proc. IEEE Engineering in Medicine and Biology Society, pp. 3296–3299, 2014.

[5] J. A. Jensen et al., "SARUS: A Synthetic Aperture Real-Time Ultrasound System," Trans. Ultrason. Ferroelectr. Freq. Control, vol. 60, no. 9, 2013.

[6] C. H. Hu, X. C. Xu, J. M. Cannata, J. T. Yen, and K. K. Shung, "Development of a real-time, high-frequency ultrasound digital beamformer for high-frequency linear array transducers," IEEE Ultrasound System," Trans. Ultrason. Ferroelectr. Freq. Control, vol. 53, no. 2, 2006.

[7] J. A. Jensen, S. I. Nikolov, K. L. Gammelmark, and M. H. Pedersen, "Synthetic aperture ultrasound imaging," Ultrasonics, vol. 44, pp. 5–15, 2006.

[8] J. Chen, J. Chen, H. Min, and X. Wang, "Real-time embedded implementation of adaptive beamforming for medical ultrasound imaging," in Proc. 6th International Conference on Instrumentation & Measurement, Computer, Communication and Control (IMCCC), Harbin, pp. 356–360, 2016.

[9] R. Y. Chiao and X. Hao "Coded excitation for diagnostic ultrasound: a system developer's perspective," IEEE Trans. Ultrason. Ferroelectr. Freq. Control, vol. 52, no. 2, 2005.

[10] P. Tortoli et al., "ULA-OP: and advanced open platform for ultrasound research," IEEE Trans. Ultrason. Ferroelectr. Freq. Control, vol. 56, no. 10, 2009.

[11] A. Assef, J. M. Maia, and E. T. Costa, “A flexible multichannel FPGA and PC-Based ultrasound system for medical imaging research: initial phantom experiments,” Res. Biomed. Eng., vol. 31, no.3, 2015.

[12] E. Boni et al., “Architecture of an ultrasound system for continuous real-time high frame rate imaging,” IEEE Trans. Ultrason. Ferroelectr. Freq. Control, vol. 64, no. 9, 2017.

[13] J. Chen, S. Zhou, and H. Min, “Implementation of parallel medical ultrasound imaging algorithm on CAPI-enabled FPGA,” in Proc. International Conference on Field-Programmable Technology, pp. 311–314, 2016.

[14] F. Angiolioni et al., “1024-channel 3D ultrasound digital beamformer in a single 5W FPGA,” in Proc. Design, Automation & Test in Europe Conference & Exhibition (DATE), Lausanne, pp. 1225–1228, 2017.

[15] M. Taher, “Accelerating scientific applications using GPUs,” in Proc. 4th International Design and Test Workshop (IDT), Riyadh, pp. 1–6, 2009.

[16] S. Che, J. W. Sheaffer, K. Skardon, and J. Lach, “Accelerating compute-intensive applications with GPUs and FPGAs,” in Proc. Symposium on Application Specific Processors, Anaheim, CA, pp. 101–107, 2008.

[17] M. Tanter and M. Fink, “Ultrafast imaging in biomedical ultrasound,” IEEE Trans. Ultrason. Ferroelectr. Freq. Control, vol. 56, no. 10, 2009.

[18] H. So, J. Chen, B. Yiu, and A. Yu, “Medical Ultrasound Imaging: To GPU or Not to GPU?,” IEEE Micro., vol. 31, no. 5, pp. 54–65, 2011.

[19] J. Chen, A. C. H. Yu, and H. K. H. So, “Design considerations of real-time adaptive beamformer for medical ultrasound research using FPGA and GPU,” in Proc. International Conference on Field-Programmable Technology, Seoul, pp. 198–205, 2012.

[20] M. Lewandowski, M. Walczak, B. Witek, P. Kulesza, and K. Sielewicz, “Modular & scalable ultrasound platform with GPU processing,” in Proc. IEEE International Ultrasonics Symposium, Dresden, pp. 1–4, 2012.

[21] J. W. Choe, A. Nikoozadeh, Ö. Oralkan, and B. T. Khuri-Yakub, “GPU-based real-time imaging software suite for medical ultrasound,” in Proc. IEEE International Ultrasonics Symposium (IUS), Prague, pp. 2057–2060, 2013.

[22] R. Kulina et al., “New software beamforming algorithm is superior to standard hardware-based beamformer in endocardial border detection,” White Paper, General Electric Company, 2016.

[23] M. Walczak, M. Lewandowski, and N. Żołek, “Optimization of real-time ultrasound PCIe data streaming and OpenCL processing for SAFT imaging,” in Proc. IEEE International Ultrasonics Symposium (IUS), Prague, pp. 2064–2067, 2013.

[24] T. I. Laakso, V. Valimaki, M. Karjalainen, and U. K. Laine, “Splitting the unit delay [FIR/all pass filters design],” IEEE Signal Process. Mag., vol. 13, no. 1, pp. 30–60, 1996.

[25] G. Matrone, A. S. Savoia, G. Caliano, and G. Magenes, “The delay multiply and sum beamforming algorithm in ultrasound B-mode medical imaging,” IEEE Trans. Med. Imag., vol. 34, no. 4, pp. 940–949, 2015.