

1024-Channel 3D Ultrasound Digital Beamformer in a Single 5W FPGA

F. Angiolini*, A. Ibrahim*, W. Simon*, A. C. Yüzügüler*, M. Arditi*, J.-P. Thiran*[†], and G. De Micheli*

* Swiss Federal Institute of Technology in Lausanne (EPFL), Switzerland

[†]Department of Radiology, University Hospital Center (CHUV) and University of Lausanne (UNIL), Switzerland

Abstract—3D ultrasound, an emerging medical imaging technique that is presently only used in hospitals, has the potential to enable breakthrough telemedicine applications, provided that its cost and power dissipation can be minimized. In this paper, we present an FPGA architecture suitable for a portable medical 3D ultrasound device. We show an optimized design for the digital part of the imager, including the delay calculation block, which is its most critical part. Our computationally efficient approach requires a single FPGA for 3D imaging, which is unprecedented. The design is scalable; a configuration supporting a 32×32-channel probe, which enables high-quality imaging, consumes only about 5W.

I. INTRODUCTION

Ultrasound (US) imaging is a prevalent medical diagnostic technique. US imaging relies on a process called *insonification*. A hand-held probe of piezoelectric elements, vibrating in the MHz range, emits acoustic waves into a body region. The echoes returned by tissue discontinuities are processed with an algorithm called *Beamforming* (BF) to reconstruct an image. Most US arrays are uni-dimensional, and insonify a planar body section, of which a 2D image can be reconstructed. A consequence is that the observation of specific tissue features requires pinpoint precision in the positioning of the probe. The probe must be handled by a highly trained sonographer, and thus, historically, US imaging has been confined to hospitals and clinics.

Another implication is that, to date, it has not been possible to decouple the US acquisitions and their diagnosis. *Telesonography* - i.e. the remote transmission of acquired scans for diagnosis - would unlock major societal benefits. Acquisitions could be even demanded to family doctors, with large cost and time savings. Telesonography would also significantly improve diagnostic options in remote regions, in emergency medicine, and in enclosed environments (ships, space stations).

Recent developments have made volumetric (3D) US possible. 3D US uses a matrix probe to insonify and image a roughly pyramidal body section. We propose to leverage 3D US to enable telesonography. The crucial point is that the much larger volume of acquisition relaxes the accuracy constraints on probe handling by the operator. For example, a single scan acquired by a minimally trained operator would suffice to capture a whole kidney. The scan could be uploaded to a remote hospital, and within it, an experienced sonographer would then proceed to diagnose. However, this is not possible today. 3D US has computational requirements orders of magnitude higher than 2D US, and as a consequence, current commercial 3D US systems are bulky, expensive and power-hungry, aimed at hospital usage [1]. Research 3D US systems are similarly complex, such as the academic SARUS [2] machine, which runs on as many as 320 FPGAs to support 1024 channels (transducer elements). The recent second iteration of the ULA-OP platform [3] uses 8 high-end FPGAs and 16 DSPs to perform 256-channel BF.

In this paper, we describe the first *single-FPGA digital 3D beamformer* that can tackle the most computationally demanding tasks of image reconstruction in a *power budget of approximately 5W for 1024 channels*. This is unprecedented, and we believe that this design represents a key component to enable a telesonography device with ubiquitous applications.

II. BEAMFORMING IN 3D ULTRASOUND

A US imager requires transmit (TX) and receive (RX) blocks. The TX block is active for a few microseconds, during which the piezoelectric elements are excited to emit an acoustic wave at a frequency dependent on probe design, usually in the range of 2-30 MHz.

After each TX phase, the acoustic wave propagates into the body and part of its energy is reflected towards the probe by scatterers, i.e. tissue inhomogeneities, in the body. The echoes received by each transducer element are then processed by the RX circuitry, which starts with analog conditioning. At this stage, especially in 3D US, the signal count is usually reduced. This is because a matrix probe can contain many thousands of elements (e.g. 9212 [1]), which is an impractically high number to condition, digitize, carry over cables from the probe to the imager, and to be processed to reconstruct an image. Multiple signals are thus analogically multiplexed or summed [4], before being digitized. Naturally, support for a higher number of *channels* (digital signal inputs) can improve resolution and other image quality parameters, but at a processing complexity cost. In this work, we aim to match the channel count of the highest-end imagers we are aware of [2], i.e. 1024.

Once the RX digital signals are available, a BF block reconstructs an image. The kernel of the BF algorithm is:

$$s(S) = \sum_{D=1}^N e(D, t_p(|\vec{OS}|) + t_p(|\vec{SD}|))w(S, D), \forall S \in V \quad (1)$$

where S is a point in the volume of interest V . The outcome $s(S)$ is a signal that follows the reflectivity of scatterers at location S , and will eventually be used to calculate the brightness of the corresponding image pixel. N is the number of input channels, while e is the amplitude of the echo received by element $D \in 1, \dots, N$ at the time sample t_p . The value of t_p represents the propagation delay that sound waves incur from a given emission reference O , to S , and back to the probe's destination element D . Echo intensities are multiplied by w to perform *apodization* [5], a process by which a window function is applied to the echo signals across the piezoelectric array during BF, in order to improve its antenna-directivity pattern. Apodization's main goal is to maximize the amplitude of the signals received from the intended direction (towards S) while minimizing that of the spurious signals received from all other angles. Equation (1) holds for both 2D and 3D imaging,

but in the latter, N and $|V|$ are larger since both the probe and the volume have an extra dimension, thus making the problem up to four orders of magnitude more challenging.

The output of this algorithm is an image that still has a frequency spectrum around the transducer's central frequency. A demodulation step follows BF in order to provide a readable amplitude-image.

In this work, we tackle the implementation of the 3D beamformer, which is the most computationally expensive part in the processing pipeline, plus the demodulation step. 3D US BF presents major computational challenges, especially for a self-contained FPGA implementation. The main ones are discussed in the following.

1) *Challenge 1: Memory Requirements for Input Samples:* Each of our supported 32×32 input channels - given our 10cm target penetration depth (to be traversed forward and backward), c , and f_s as specified in Table I - produces 4156 data samples per insonification. Common ADC precisions for US applications are 12-16 bits; assuming 2 bytes per sample, over 8 MB of samples are produced per insonification. This dataset is beyond what can be practically stored on-chip in most architectures, forcing a streaming design that reconstructs slices of the image from small subsets of the input data.

2) *Challenge 2: Processing Rate:* Equation (1) requires highly iterative computations. Both the delay t_p and the apodization w functions must be evaluated $|V| \times N$ times to reconstruct a volume. For our target specifications in Table I, this equates to 2.5 billion evaluations per frame. At a desired frame rate of e.g. 30 volumes per second (vps), 75 billion evaluations per second are required. Unfortunately, the delay calculation implies solving a square root to retrieve the Euclidean distance between S and D , while the apodization process requires the evaluation of a windowing - such as, a Hanning - function and then a multiplication. Evaluating these complex functions in realtime is simply infeasible within a single FPGA.

3) *Challenge 3: Memory Requirements and Bandwidth for Delay and Apodization Functions:* An intuitive solution to the challenge above is to pre-calculate the delay and apodization functions for all possible geometries, and then to fetch the values on the fly from tables. Based on the same reasoning above, 2.5 billion distinct delay and as many apodization values would need to be stored; assuming a 16-bit representation of the values, this equals 10 GB of coefficient tables, necessarily stored off-chip. All 10 GB of coefficients would then need to be fetched 30 times per second, for a bandwidth of 300 GB/s. This implies unrealistic hardware costs and power budgets for the memory subsystem¹.

4) *Challenge 4: Delay Calculation Accuracy:* The quality of the reconstructed images depends heavily on the accuracy of the beamformer calculations. Delay calculation imprecisions may result in poor focus, image artifacts, aliasing, etc.

In conclusion, a brute-force approach to 3D US imaging is incompatible with a portable, affordable device that must comprise limited processing resources. Instead, approximate algorithms and optimized architectures must be deployed. Their accuracy and image quality then need to be assessed.

III. PREVIOUS WORK

Telesonography could be enabled by 3D US, where an untrained operator can scan a complete volume and capture small features easily. Unfortunately, today's 3D machines are

¹Note that, in a special fully symmetric geometry setup, leveraging quadrant symmetry can reduce the memory footprint requirements by 4X, but this solution is not generic and the bandwidth constraint may still hold, depending on implementation details.

expensive and stationary [1]. This is explained by the high computational demands of 3D reconstruction algorithms. In fact, to manage the hardware costs, even state-of-the-art 3D US systems are forced to decrease the number of channels that are carried from the transducer head from a few thousands to a few hundreds [4], [6] using micro-beamforming, multiplexing, or sparse 2D arrays. Pre-beamforming makes the digital BF orders of magnitude less complex, but it limits the image quality. It is desirable to have a fully-digital, high-channel-count BF. However, the large amount of input signals to be processed individually still poses a major computation and bandwidth challenge [2]. In particular, the problem of how to compute delay coefficients to feed such beamformers at a very high throughput has been recognized as critical. For example, Sonic Millip3De [7] implements ultra-fast imaging for 128x96 transducer elements (of which only 1024 are considered per shot) with a powerful die-stacked package. Its main bottleneck is that it requires an external DRAM memory to store BF delay coefficients, and several GB/s of memory bandwidth. Other works [2], [8] have shown that a feasible alternative is to compute all delays on-the-fly on-chip, but the challenge of supporting a high-channel-count realtime 3D BF remains daunting.

In this paper, we present an innovative single-FPGA 3D US beamformer supporting 1024 channels in a 5W power envelope. To tackle the challenges described previously, we rely on a variety of approaches and architectural optimizations. This paper builds upon our previous work [9], which proposed exclusively an approximation algorithm for delay calculation, and extends it significantly by (i) optimizing the delay calculation itself to remove the need for high off-chip memory bandwidth, (ii) considering all the other components of 3D BF, such as sample storage and apodization, (iii) adding circuitry for US image demodulation. We target our beamformer to the Kintex UltraScale KU040 FPGA [10], which is a latest-generation, but mid-range, part. This choice offers an optimal trade-off between FPGA capacity, price, low power consumption, and easy availability.

IV. PROPOSED SINGLE-FPGA 3D BEAMFORMER

A. Nappe-by-Nappe Processing

To tackle Challenge 1, we resort to a *nappe-by-nappe* beamformer [8]. While traditional architectures reconstruct images one radial line at a time, this reconstruction sequence resembles successive onion layers. The key advantage is that each nappe is reconstructed from temporally thin contiguous inputs or *blankets*, which can be stored on-chip. Blanket thickness depends on geometric variables and varies with depth; with the parameters in Table I, the thickest blanket fits in just 306 kB of memory, which is easily affordable. We store the input samples onto Xilinx BRAM memory blocks, which can hold 1024 rows of 18 bits. To improve beamformer throughput and since Xilinx Ultrascale BRAMs support a

TABLE I
SYSTEM SPECIFICATIONS

Parameter	Symbol	Value
Speed of sound in tissue	c	1540 m/s
Transducer center frequency	f_c	4 MHz
Transducer matrix size	$e_x \times e_y$	32×32
Element directivity (acceptance angle)		0.707 rad
Imaging volume ($\theta \times \phi \times d_p$)		$73^\circ \times 73^\circ \times 10\text{cm}$
Sampling frequency	f_s	32 MHz
Focal points	$ V $	$64 \times 64 \times 600$
Insonification origin	O	(0, 0, 0)

maximum of two concurrent reads, we map the input samples from two channels onto each BRAM. So, 512 BRAMs are required.

B. Delay “Steering”

The TX delay $t_p(|\vec{OS}|)$ is calculated in full, since it depends only on $S = (r, \theta, \phi)$, while O is fixed. A total of $2.5M$ delays must be calculated per volume, which is easily feasible.

The RX delay $t_p(|\vec{SD}|)$ is much more critical, since it must be evaluated for a varying scatterer location S and transducer element $D = (x_D, y_D, 0)$, i.e. about 2.5 billion times per volume. To perform the delay calculation efficiently in realtime (Challenge 2), we approximate [9] the square root function by means of a first-order Taylor expansion. This simplifies the calculation to:

$$t_p(|\vec{SD}|) \approx t_p(|\vec{RD}|) - \frac{x_D \sin \theta}{c} - \frac{y_D \sin \phi \cos \theta}{c} \quad (2)$$

Thanks to the proposed Taylor approximation, the RX calculation is simplified into that of a *reference delay* for a set of points $R = (0, 0, r)$ along the central line of sight (where $|R| \times |D| = 600 \times 32 \times 32 \approx 614k$; mapped on a Xilinx CORDIC core) followed by a *steering* operation, consisting of two additions. The steering coefficients depend on the speed of sound c , which is assumed to be constant at 1540 m/s, and the geometry variables x_D, y_D, θ, ϕ . Since each of these variables can only take limited and discrete values, the steering coefficients can either be precalculated or calculated on the fly, solving Challenge 3.

The implemented design is shown in Fig. 1(a). This design is a significant improvement over [9], since the TX delay and the delay coefficients are stored or calculated entirely in-FPGA, removing the bandwidth cost of fetching them from the off-chip memory. The delay calculation logic produces 32×32 delay values per cycle, used to address all 512 BRAMs simultaneously with two read addresses per BRAM, thus selecting one input sample for each probe element. The resulting 1024 samples are fed into a pipelined adder tree, which realizes the summation of (1), for an overall throughput of one reconstructed voxel per cycle (Fig. 1(b)).

C. Static Apodization

A fully general apodization implementation $w(S, D)$ according to (1) would not fit into a single FPGA. However, commonly-used expanding-aperture apodization windows [5] tend to converge to a function $w(D)$ that depends only on D after a certain distance from the probe, which is just 1cm in our setup. We thus choose to fall back on a simpler *static apodization* approach which uses this $w(D)$ (a Hanning window) over the whole volume. In total, only $|D| = 1024$ apodization values are needed; these are pre-computed and stored in a single BRAM. We apply this simplified apodization as a constant weight to each transducer element when loading the input element samples into the beamformer BRAMs (Fig. 1(b)). This reduces implementation costs greatly (Challenge 3) and affects image quality only slightly and only in the very shallowest region of the image.

D. Demodulation

The beamformed image is still in Radio Frequency (RF) and must be demodulated before visualization. In our platform, to optimize resources, we adopt the simplest available technique, which consists of calculating the absolute value of the RF image and then applying a low-pass filter implemented as

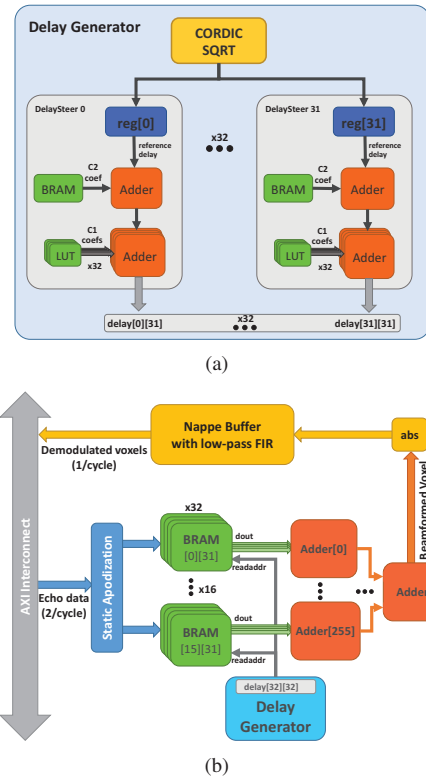


Fig. 1. Proposed highly-parallel streaming beamformer architecture. (a) TX and reference RX delays are computed and steering coefficients are applied, thus calculating 32×32 delays that are used to index the input sample BRAMs. (b) These samples, which are statically pre-apodized, are summed to reconstruct a voxel, then demodulated.

an FIR filter. Every time a voxel is beamformed, i.e. once per cycle, the absolute value of its brightness is calculated (Fig. 1(b)). This value is then stored into a circular buffer that can hold a parametric number P of nappes. A low-pass $(P - 1)$ th-order FIR filter is run on P buffered voxels; the filtering is thus along the lines of sight of the volume. The filtered outcome is stored into a further temporary buffer, from which it can be sent off-chip for visualization. We assume a filter depth of $P = 5$ nappes when evaluating image quality and resource utilization.

E. Overall System Architecture and Testing Flow

The overall FPGA architecture includes a Xilinx Microblaze with dedicated on-chip memories and peripherals. Intra-FPGA communication occurs over a 32-bit AXI interconnect, to which the proposed beamformer block is also connected. The beamformer is written in a mix of Verilog and VHDL.

The Microblaze is in charge of supervising the platform operations. An Ethernet controller is used for all communication with the external world. On the input side, data that represents the digitized transducer signals is sent over the Ethernet port, and the Microblaze loads them in blankets into the beamformer’s input BRAMs. The Microblaze reads back demodulated nappes of voxels from the beamformer and pushes them out through the Ethernet port. A laptop running a custom Visual C# application feeds the board with inputs and visualizes its outputs.

Currently, we do not have access to a matrix probe with open specifications to connect to our beamformer for real-time

imaging. Therefore, we are using simulated data. This US data is generated in Matlab via the Field II package [11]. Further, we developed Matlab scripts to produce a set of files, e.g. constants and initialization data, that are used to synthesize the FPGA design. These files are loaded into the beamformer either at synthesis time, via include files, or at FPGA boot time, via BRAM initialization in the bitstream.

V. EXPERIMENTAL RESULTS

A. FPGA Implementation Results

Table II shows that we can fit a complete 32×32 -channel beamformer on a single Kintex UltraScale KU040 FPGA [10]. The most critical resource utilization is that of the BRAM blocks; out of 1200 in the chip, 71% are used, notably 512 to store input samples and another 256 for delay calculation coefficients. The beamformer actually requires only about 30% of the FPGA lookup tables and 15% of the flip-flops, the remaining occupation being attributable to the Xilinx infrastructure IP. The architecture can scale to a larger channel count, but it is limited on the KU040 by the BRAM resources. On a larger FPGA, like the Virtex UltraScale XCVU190 [10], the channel count could be increased to 90×90 , which is about 9 times higher than commercial and research implementations [2].

The design meets timing at 133 MHz for the whole AXI subsystem, including the beamformer. This translates into an excellent *potential* sustained volume rate of over 53 volumes per second (vps), given that each volume comprises 2.5 million voxels and the throughput is one voxel per cycle. However, the volume rate presently achievable by our demonstration setup is less than 1 vps, due to the bottleneck of the Gigabit Ethernet interface to stream both inputs and outputs, its protocol overhead, and the Microblaze overhead due to moving data across the FPGA. Nonetheless, we do not regard these bottlenecks as significant, as they are contingent to the demonstration setup. An actual imaging device would feature a direct connection between the probe's ADCs and the input of the beamformer logic.

The design has an estimated power consumption of 5W, which is fully compatible with deployment into a portable, battery-operated device.

B. Delay Inaccuracy Quantification

Since the proposed architecture minimizes hardware cost by way of approximations, it is important to assess the quality of the reconstruction. A full analysis of the reconstruction quality has been performed on Matlab, but cannot be presented here for space reasons. Fig. 2 [12] shows the spatial distribution of the delay calculation inaccuracy; it can be seen that the inaccuracy is only significant in peripheral areas of the image, which are usually not critical for diagnosis. Thus, Challenge 4 is satisfactorily tackled.

VI. CONCLUSIONS

In this work we have developed a design that is able to perform single-chip BF for real-time 3D US. In a Kintex

TABLE II
BEAMFORMER ARCHITECTURE RESULTS.
*Kintex UltraScale KU040 implementation results.
**Virtex UltraScale XCVU190 extrapolated results.

Supported Channels	LUTs	Regs	BRAM	DSP	Clock	Volume Rate
32×32 *	49.6%	24.1%	71.0%	2.2%	133MHz	53.2vps
90×90 **	65.0%	33.3%	91.2%	2.3%	133MHz	53.2vps

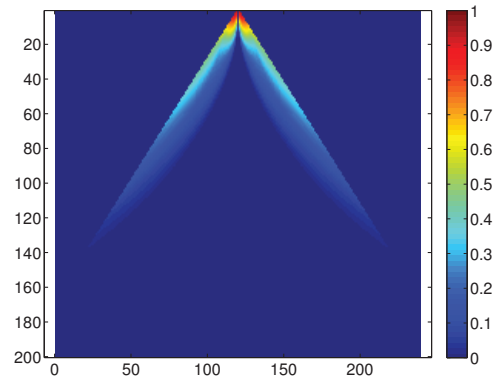


Fig. 2. For each voxel S on the XZ plane of the image, this plot shows the percentage of elements that incur a delay inaccuracy greater than 2 samples, which represents a noisy contribution. The inaccuracy is confined to the edges and the very shallow region (1-2cm) of the image.

FPGA, it supports a matrix probe with 1024 channels, and achieves a theoretical frame rate of 53 fps, with an estimated consumption of 5W. This makes our design suitable for a portable, battery-operated 3D US device that can enable teleonography, with major potential societal benefits.

ACKNOWLEDGMENT

The authors would like to acknowledge funding from the Swiss Confederation through the UltrasoundToGo project of the Nano-Tera.ch initiative.

REFERENCES

- [1] Philips Electronics N.V., "Philips iU22 ultrasound with xMATRIX system specifications," 2012, www.healthcare.philips.com.
- [2] J. Jensen, H. Holten-Lund, R. Nilsson, M. Hansen, U. Larsen, R. Domsten, B. Tomov, M. Stuart, S. Nikolov, M. Pihl, Y. Du, J. Rasmussen, and M. Rasmussen, "SARUS: A synthetic aperture real-time ultrasound system," *Ultrasonics, Ferroelectrics, and Frequency Control, IEEE Transactions on*, vol. 60, no. 9, pp. 1838–1852, Sep 2013.
- [3] E. Boni, L. Bassi, A. Dallai, F. Guidi, V. Meacci, A. Ramalli, S. Ricci, and P. Tortoli, "ULA-OP 256: A 256-channel open scanner for development and real-time implementation of new ultrasound methods," *Ultrasonics, Ferroelectrics, and Frequency Control, IEEE Transactions on*, in press 2016.
- [4] B. Savord and R. Solomon, "Fully sampled matrix transducer for real time 3D ultrasonic imaging," in *Ultrasonics, 2003 IEEE Symposium on*, vol. 1. IEEE, 2003, pp. 945–953.
- [5] C. Daft and W. Engeler, "Windowing of wide-band ultrasound transducers," in *Ultrasonics Symposium, 1996. Proceedings., 1996 IEEE*, vol. 2. IEEE, 1996, pp. 1541–1544.
- [6] J. Um, Y. Kim, S. Cho, M. Chae, J. Song, B. Kim, S. Lee, J. Bang, Y. Kim, K. Cho, B. Kim, J. Sim, and H. Park, "An analog-digital hybrid RX beamformer chip with non-uniform sampling for ultrasound medical imaging with 2D CMUT array," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 8, no. 6, December 2014.
- [7] R. Sampson, M. Yang, S. Wei, R. Jintamethasawat, B. Fowlkes, O. Kripfgans, C. Chakrabarti, and T. Wenisch, "FPGA implementation of low-power 3D ultrasound beamformer," in *2015 IEEE International Ultrasonics Symposium (IUS 2015)*, Nov 2015.
- [8] P. Vogel, A. Bartolini, and L. Benini, "Efficient parallel beamforming for 3D ultrasound imaging," in *Proceedings of the 24th Edition of the Great Lakes Symposium on VLSI (GLSVLSI '14)*, 2014, pp. 175–180.
- [9] A. Ibrahim, P. A. Hager, A. Bartolini, F. Angiolini, M. Arditi, L. Benini, and G. De Micheli, "Tackling the bottleneck of delay tables in 3d ultrasound imaging," in *Proceedings of the 2015 Design Automation and Test in Europe (DATE 2015) Conference*, March 2015, pp. 1683 – 1688.
- [10] Xilinx Inc., "Ultrascale architecture," 2014, www.xilinx.com/products/technology/ultrascale.html.
- [11] J. A. Jensen, "Field: A program for simulating ultrasound systems," in *IEEE Symposium on Biomedical Imaging*, vol. 4, 1996, pp. 351–353.
- [12] A. Ibrahim, F. Angiolini, M. Arditi, J.-P. Thiran, and G. De Micheli, "Apodization scheme for hardware-efficient beamformer," 2016.