DOI: 10.61782/fa.2025.0583

FORUM ACUSTICUM EURONOISE 2025

MODULAR AND FLEXIBLE HARDWARE ARCHITECTURE FOR REAL-TIME 3D ULTRASOUND IMAGING

Cruza, Jorge F.[1*] **Mateos, Raul**[2] **Bueno, Ricardo**[3]
Montaldo, Gabriel[4] **Camacho, Jorge**[1]
[1] Department of Sensors and Ultrasonics Systems, ITEFI (CSIC), Spain
[2] Department of Electronics, University of Alcalá, Spain
[3]DASEL SL Avda del Cañal 44, Nave 3 - 28500 Arganda del Rey (Madrid), Spain
[4]Neuro-Electronics Research Flanders, Leuven, Belgium

ABSTRACT

Ultrasound 3D imaging with arrays usually requires hardware systems with more active channels than for conventional 2D imaging, which increases systems complexity. Furthermore, it increases the beamforming difficulty, as more image lines have to be generated (3D volume) using more acquired raw signals (A-Scans). Typical approaches to this problem can be divided in hardware and software solutions. In the first, the beamforming process is carried out by a specialized hardware (FPGA, ASIC's, etc.), with lower power consumption, but also with lower flexibility. On the other hand, software beamformers are very flexible in terms of programming, but they require a much larger bandwidth with the acquisition system. This work presents a modular and flexible hardware architecture for 2D and 3D ultrasound imaging, which combines hardware beamforming capabilities with high-speed links with a processing computer. This approach enables to work with both hardware and software beamformers, taking full advantage of each one of them. A description of the architecture is given, along with the expected performance in terms of hardware processing power and data transfer rate. Finally, application examples are given: Total Focusing Method imaging in non-destructive-testing, and functional ultrasound imaging in the brain for pre-clinical models.

Keywords: *ultrasound, beamforming, 3D imaging*

1. INTRODUCTION

Ultrasound systems have evolved in the last years from hardware processing with highly specialized electronics, to software beamforming by high capacity computers, mainly because of the availability of powerful Graphical Processor Units (GPU), which are very efficient for the implementation of typical ultrasound imaging algorithms, and are easy to develop with than hardware architectures. Nevertheless, the software approach has limitations with regard to power consumption and integration for the development of industrial and medical devices, which prevents to completely abandon the specialized hardware approach.

This work presents a flexible and modular architecture able to fulfill the requirements of both, hardware and software beamformers. The idea behind that is to provide a platform were new developments can be performed faster and at lower cost by software programming, mainly oriented to scientific research and early stages of final product R&D, while it still have the capacity of implementing the more demanding parts in hardware, at higher processing rates with less power consumption, for final deployment.

A description of the architecture is presented, along with practical examples in two application fields: pre-clinical imaging and non-destructive-testing.

[*]***Corresponding author:*** *jorge.f.cruza@csic.es*

2. HARDWARE ARCHITECTURE

The philosophy of this system is a flexible architecture to work in two different scenarios:

1. Hardware beamforming (HWBF): Whole beamforming and post-procesing is performed by the ultrasonic system and the host PC just displays the image.
2. Software beamforming (SWBF): System sends in real time the raw acquisition data in order to be processed by a high performance computer.

The architecture of the system is scalable, in order to build systems from 32 channels to 512 active channels, keeping the ability of real time beamforming the image in real time.

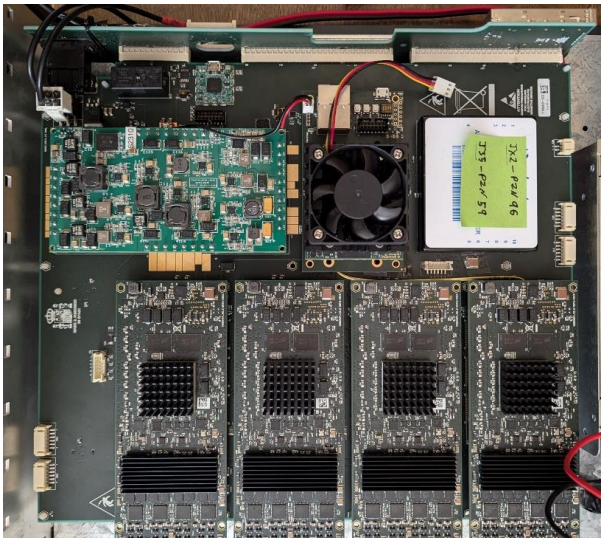

Figure 1. Motherboard of a 128 channels system, with 4x32 channels USMOD32 modules and FPGA SoC control unit.

2.1 System architecture

The control of the emission circuits, the digitalization of the received signals and the beamforming of the image are carried out in a distributed way in the system's FPGAs, transferring the final image to a host computer. The system also has the enough output bandwidth to send raw data to a remote system to be processed in real time. An FPGA SoC with embedded ARM processors performs the control of the system, and also implements the algorithms that are not too intensive but whose execution on the host PC can be a disadvantage due to larger latency.

The main design goals of the architecture are scalability, distributed processing and flexibility. Scalability is important to cover applications that require different numbers of channels, which is common in NDT. On the other hand, distributed processing is essential to handle the large volume of generated data. The third goal is flexibility, i.e. to be able to easily implement different imaging techniques, which is not only achieved by the reprogrammable nature of the FPGAs, but also by being able to evacuate the raw data in real time to an external processing system, e.g. a high-performance PC with GPUs.

The system is based on modules (USMOD32) capable of controlling, digitalizing and processing 32 ultrasound channels. The modules are connected together via a high-speed segmented bus to make possible systems with any number of channels in blocks of 32. **Figure 1** shows an implementation with 128-channels. Each module contains the pulsers, protection circuits and analog front-end (amplifier, filter and A/D converter) for 32 channels and 1 GB of DDR memory to store the beamforming results or the received signals. The selected FPGA is the XC7K160T, from Advanced Micro Devices (AMD, formerly Xilinx) Kintex series. The footprint is fully compatible with the higher-capacity XC7K325T and XC7K410T models for systems that require more hardware processing power.

Two segmented buses connect the FPGAs of the processing modules. The first, based on 10 LVDS pairs (1 Gb/s per pair), is used for programming acquisition and processing parameters and sending the images already conformed to the FPGA-SoC, which can carry out additional processing prior to sending them to the host PC.

The second bus is based on AMD GTX 7-series transceivers, which provide up to 10 Gbps full duplex communication in each direction. The FPGAs integrate 8 GTX transceivers, that are used to connect each module with the previous one with 4 GTX and to the next module with the remaining 4. Thus, closing the circuit (**Figure 2**) results in two circular buses of 40 Gbps each, with data flowing in opposite directions, allowing data transfer between any two FPGAs in the system at 80 Gb/s if both directions are used. This high-speed bus is used to transmit the partially beamformed images between the FPGAs of the modules, since each image line requires information from all channels.

In the HWBF mode, each module beamforms the image lines with the information acquired by its 32 channels, and sends them over one of the two 40 Gbps buses to the next FPGA. This one adds the results of beamforming its own channels with the data received from the previous one and sends the result to the next module.

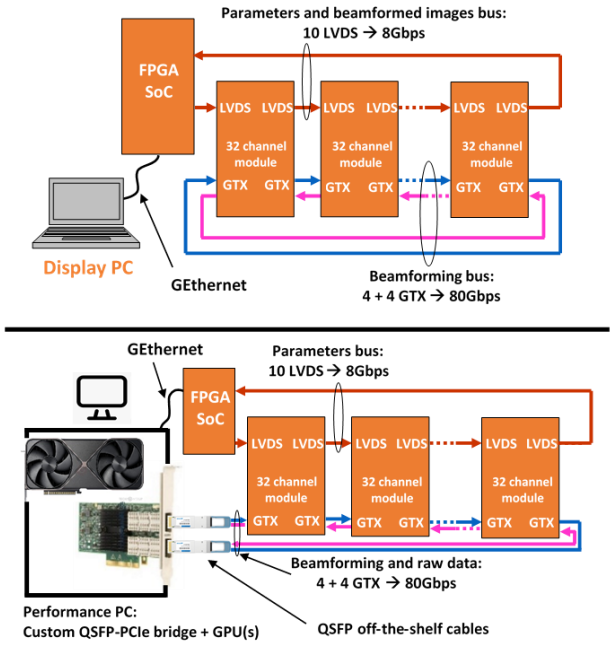

Figure 2. *Top*: configuration for hardware FPGA beamforming, host PC only for display. *Bottom*: configuration with a high performance PC, where beamforming process can be done by software.

When the data has travelled through all the FPGAs, the image is fully formed.

The image is stored in the dynamic memory (DDR), and it is divided into as many parts as USMOD32 are present in the system. This leverages not only the processing power of all the FPGAs, but also the memory bandwidth of all the modules, making the system a high-speed distributed processing architecture. This is the main difference with regard to previously proposed centralized architectures, where a central node requires a large memory bandwidth to store the result. This is particularly important in TFM and PWI image modes, since this kind of image modes have to sum many sub-images obtained with different emission, previous results has to be loaded from memory, summed with current ones and stored back to memory.

For the SWBF, the high bandwidth bus is used to send raw acquisition previously stored in the DDR of the USMOD32 through 2 QSFP cables. These cables are connected to a custom FPGA design implemented in a FPGA based PCIe board with 2 QSFP sockets, mounted in 8-lane PCIe 3.0 slot in a PC. The data from ultrasound system is stored in the PCI board DDR, and then is sent via DMA to PC main memory for the software processing of the ultrasonic data.

2.2 FPGA architecture

Figure 3 shows the representation of the receiving datapath. Other necessary subsystems, such as the emission beamformer, have been omitted in the figure for clarity. In general, the connections shown in this image follow the AXI4-Stream protocol, except for the connection to the DDR which is AXI4 and the 'register access bus' which is a 'custom' register access protocol similar to AXI4-Lite [1]. For the sake of clarity, the data width and operating frequencies of the buses have not been made explicit, but as a guide, they have been drawn with the width in relation to the total bus capacity. For example, the wider buses are capable simultaneously acquiring 32 signals and transmitting the beamforming result, therefore, they have a capacity of at least 28Gb/s. The narrower busses are sized to be transmitted over the LVDS links and, therefore, they have a maximum capacity of 8 Gbps (typically 32 bits at 250 MHz). The element named 'AXI-infrastructure' in Figure 3 consists of several elements of varying complexity such as multiplexers, DMAs, switches, clock domain changes, to name a few.

The datapath starts with the signals of the 32 channels being amplified, filtered and digitalized by the Analog Front End (AFE). Once inside the FPGA, the signals are filtered with a symmetrical FIR filter with 63 coefficients, followed by optional decimation. Afterwards, an adder with a high-capacity FIFO can perform averaging of up to 16 acquisitions. At this point, the acquisition can take two paths: it can either go into the beamformer or it can be stored in the module's DDR as raw data.

In the case of a remote PC performs the beamforming, the connection shown in red is used to dump the data from the DDR to the QSFP links. The protocol used to connect both the FPGAs to each other and the FPGAs to the PCIe card is the AURORA 64/66 protocol [2].

In case of hardware beamforming, the acquired signals can be input directly from the output of the averaged signals or can be loaded from the DDR memory. This is particularly useful if the same acquisition is to be beamformed several times by modifying the parameters (Multi-beamforming).

The beamformed data is summed with that of the precedent FPGAs, which are received from the GTX links. The sum is forwarded to the next FPGA, however, part of this sum has to be accumulated to the previous sub-images. Each FPGA is allocated a part of the image, in order to distribute the memory bandwidth usage among all modules.

2.3 Hardware beamforming

The beamforming operation is performed by a delay sum beamformer with a novel architecture previously developed

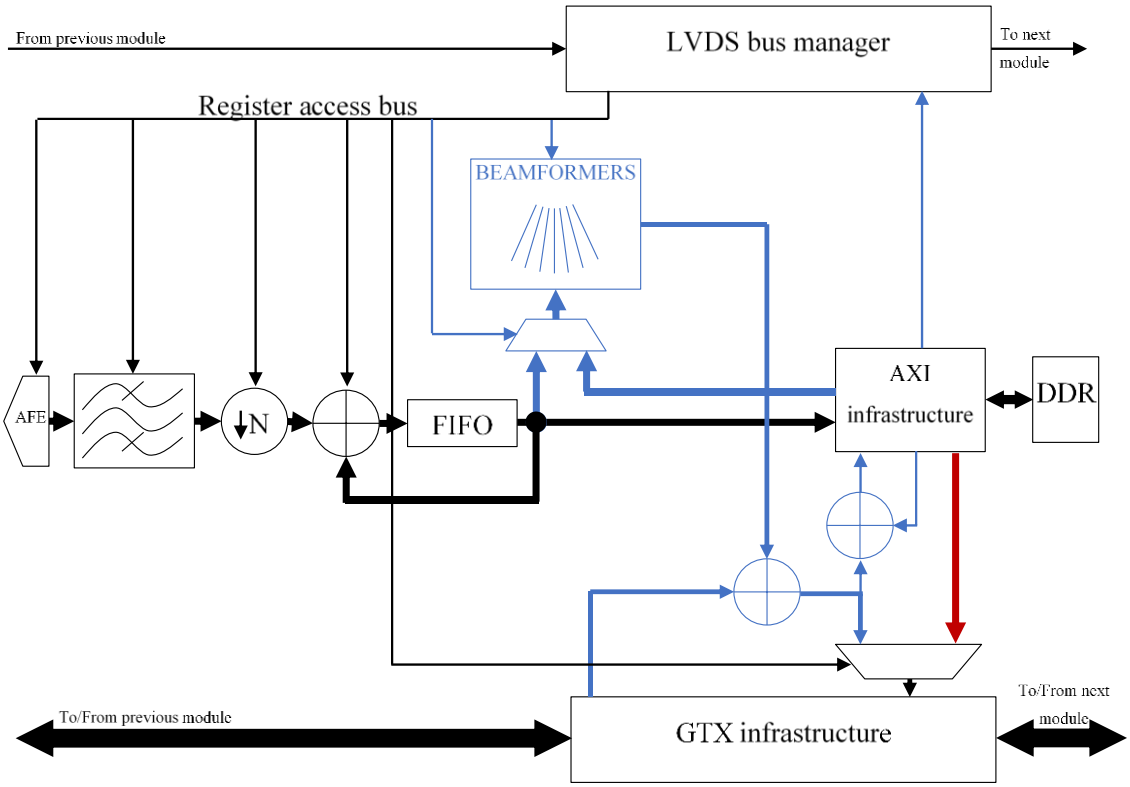

Figure 3. Schematic representation of the datapath inside each FPGA. Elements in black are utilized in both hardware and software beamforming. Blue elements are necessary for hardware beamforming while red connection is only used to dump raw acquisitions to the PC for a software beamforming.

by our group [3]. The delays are performed by using fractional delay filters [4] in order to get time resolution above the sampling frequency [5]. This design features a systolic and parallel architecture for interpolation and beamforming where fractional filtering and the process of sum are imbricated in the same hardware. Besides, this architecture makes an extensive use of the DSP tiles present in FPGAs [6]. These tiles are "hardwired" inside FPGA and therefore are less power consuming and can switch at higher frequencies than circuits implemented with regular FPGA resources, allowing time-multiplexing of these resources. Another key of the design are the delay calculators. Unlike other beamformers [7] where the delays are pre-calculated and loaded in FPGA memory, this beamformers uses a circuit to calculate iteratively the time delay that only needs parameters with a total of 80 bits wide. The delay calculator circuit is a pipelined version of the presented in [8] and a key part of the patent [9]. The pipelining, allows this logic switch at 250MHz using FPGA fabric resources.

Figure 4 shows the schematic of the beamformer. The samples coming from ADCs are stored in a 16 bit wide, 2048 deep circular buffer (one 32Kb Block RAM per channel). Simultaneously, the delay calculator output is used in this way: the integer part of the delay corresponds to index of the sample that is the input of the fractional filter, and the fractional part of the delay, chooses the set of coefficients used to apply the desired fractional delay (0, 1/4, 2/4 or 3/4). The delayed sample is automatically summed to the delayed sample of the other channels since delay filters are cascaded. Whole structure switches at 250MHz, so, we can time multiplex it in order to calculate more than one scan line at once. For example, with a broadband array of 15MHz center frequency where the sampling frequency can be 62.5 MHz, 4 scan lines can be obtained in parallel, while, in a 3MHz center frequency with 15.125MHz sampling frequency, 16 scan lines can simultaneously obtained. This can be useful in the conventional phased array technique if the emission focus is broad enough to beamform multiple scan lines of the image

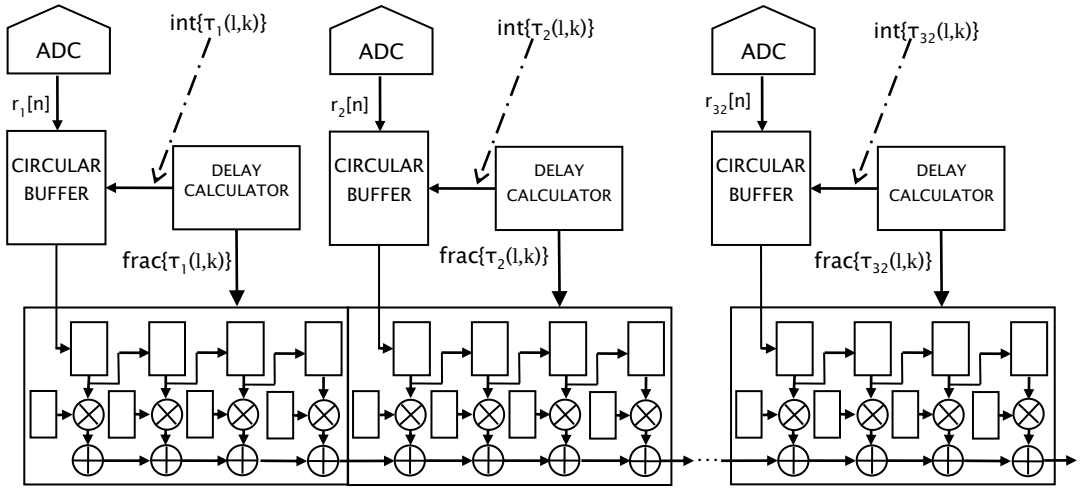

Figure 4. Beamformer architecture: fractional delay and the sum are performed by same FPGA resources, the DSP tiles.

in every emission. But is even much more interesting in plane wave image (PWI) or total focussing method (TFM) where emission is unfocussed and whole image have to be beamformed after every emission event (sometimes called "sub-image"). The ability of implement PWI or TFM in real time depends in the capacity to beamform all scan lines of the sub-image before next emission event. In this work we want to analyze three examples:

A PWI for functional ultrasound (fUS) for rat brain: c=1.5 mm/us, 15MHz center frequency (F_s=62.5 MSPS), L=128 lines of d=15mm depth, 500 beamformed images per second, each one compounded by 7 plane waves (PRF=3500Hz).

B Total Focusing Method: steel c=6.25 mm/us, 3MHz center frequency (15.125MSPS), 128 lines of 50mm depth, 25 images/s compounded by 128 sub-images, one per array element (PRF=1920Hz).

C 3D Total Focusing Method: 16x16 array (256 elements), steel c=6.25 mm/us, 5MHz center frequency (31.25MSPS), 256 lines of 50mm depth, 25 volumes/s compounded by 256 sub volumes, one per array element (PRF=6400Hz).

To determinate if the hardware is able to implement these examples in real time we have to define the next figures: $\mathbf{T_{PRF}}$ (1 / PRF) is the between different acquisitions. $\mathbf{T_A}$ is the time it takes acquiring necessary samples for the image, **L** is number of image lines, **M** is the relationship between switching frequency of the beamformer and acquisition sampling frequency and defines the number of lines in parallel can process in the acquisition time ($\mathbf{T_a}$). **N** is Number of beamformers (the structure shown in Figure 3 can be replicated inside FPGA). **R** is the times the beamformers have to reprocess the same acquisition data in order to beamform all the L lines of the image.

$$R = \left\lceil \frac{L}{M \cdot N} \right\rceil$$

Furthermore, to operate in strict real time, the time that takes to beamform the image (or sub-image) has to be lower than the time between different acquisitions:

$$R_{MAX} = \left\lfloor \frac{T_{PRF}}{T_A} \right\rfloor$$

Using these definitions, **Table 1** shows the number of beamformers needed for the three examples presented, employing this equation:

$$N = \left\lceil \frac{L}{M \cdot R_{MAX}} \right\rceil = \left\lceil \frac{L}{M \cdot \lfloor T_{PRF}/T_A \rfloor} \right\rceil$$

As **Table 1** reveals, the more exigent image mode, the more beamformer structures that have to fit in the FPGA to operate in real time. **Table 2** shows the resource used for the beamforming, taking into account that there are also other subsystems that have to fit in FPGA. Furthermore, the total resource consumption should be below 80% in some critical resources (LUTs

particularly) in order to allow place-and-route tool to reach timing closure.

Table 2~~2~~1. Number of beamformers needed for each acquisition mode

	A	B	C
c (mm/µs)	1,5	6,25	6,25
D (mm)	15	50	50
Ta (µs)	20	16	16
L (lines)	128	128	256
M (lines/beamformer)	4	16	8
FPS (images/s)	500	25	25
Sub-images (angles or elements)	7	128	256
PRF (Hz)	3500	3200	6400
T_{PRF} (µs)	285,71	312,50	156,25
Rmax	15	19	9
N (beamformers)	**3**	**1**	**4**

As can be appreciated with **Table 1** and **Table 2**, the mid-range FPGA XC7K160 can implement complex types of image like example B, but is unable to perform fUS in real time (example A)[10]. However, a system which features high range FPGAs can perform fUS or 3D TFM in strict real time without need a powerful external computer. Despite high-end FPGAs have a higher cost, their lower power consumption and size than a solution based on an external PC, could justify their use for industrial developments.

3. RESULTS

Three application examples with a 128 channels system developed with the proposed architecture are presented. **Figure 5** shows a functional ultrasound image of a rat brain, obtained with a 15 MHZ phased-array transducer, while a visual stimulus is applied to the animal. The Doppler image was generated by the ultrafast plane-wave technique [10] with 5 angles between -6º and 6º, 2 emitted pulses, 8 kHz PRF, 4 averages and a total of 8.000 consecutive images (20 seconds continuous acquisition). The output trigger of the system was received by the visual stimulus system and delayed 10 seconds before emitting a vertical lines pattern to the right eye of the mouse. This way, a base-line reference of 10 seconds can be compared with a stimulus of 10 seconds to detect the activated brain zones. The experiment was repeated 10 times and the response was averaged for better signal-to-noise ratio. Figure 8 shows the functional image in color, overprinted on a single Doppler image in black and white. The activated zone corresponds with that expected for the given stimulus, and the spatial resolution and SNR are in accordance with the experiment set-up.

A second experiment is presented, in this case, for a non-destructive-testing application (**Figure 6**). An 11x11 matrix array of 3.5 MHz was used for imaging inside a calibration block with several flat-bottom holes at different depths. The virtual array method was used [11] to obtain, in real time, the focusing delays inside the material, accounting for the refraction at the interface. The high-speed links were used to transfer data to a PC with an NVidia RTX4080 GPU, where the beamforming algorithms achieved an imaging rate of 10 volumes per second, automatically detecting the component surface location and shape (3D auto-focus).

Figure 7 shows the HWBF performance of the system, by acquiring a TFM image with a 15 MHz 128 elements array, in real-time. The beamformed region is 36mm wide (probe has an aperture of 32 mm) and 50mm depth. The beamformed image size has 256 lines wide and 700 points per line depth. With this configuration, it took 230ms to generate that image (4 frames per second). The system used for this experiment ships the smallest compatible FPGA (XC7K160), so this frame rate can be multiplied by 4 using higher range FPGAs.

Table 3~~3~~2. Number of beamformers that can fit in the 3 different pin compatible FPGAs. In bold, the limiting resource.

	one beamformer	XC7K160		XC7K325		XC7K410	
# BEAMFORMERS	1	2		4		8	
LUTs	10778	21556	21,26%	43112	21,15%	86224	33,92%
64-bit distributed RAM	6149	12298	35,13%	24596	38,43%	49192	**54,29%**
36Kb Block RAM	32	64	19,69%	128	28,76%	256	32,20%
DSPs	128	388	**64,67%**	512	**60,95%**	1024	**66,49%**

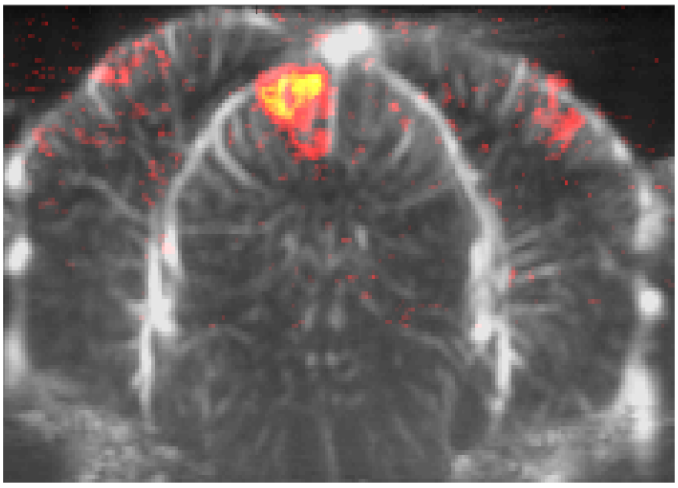

Figure 5. Hardware beamformed TFM with imasonic 128 elements 15MHz center frequency, 0.25 pitch probe, 36x50mm, 256 x 750pixels.

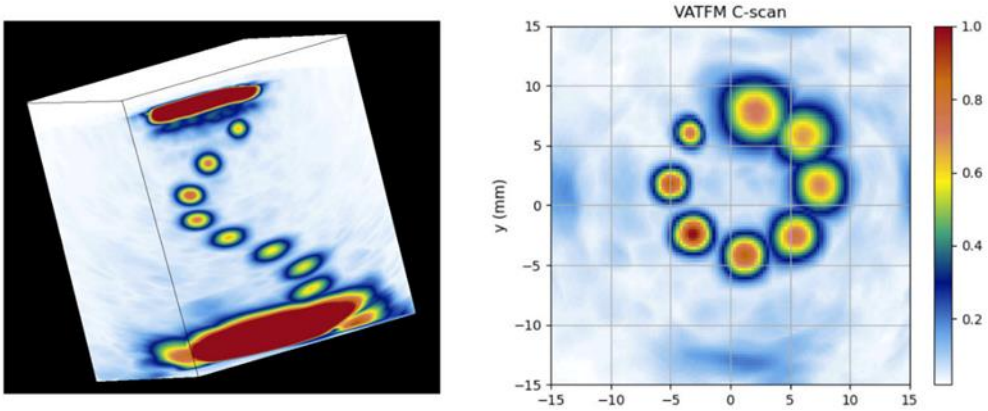

Figure 6. Left: 11x11 matrix array TFM image3D image SW processed with CUDA. Rigth: C-scan of the image.

Figure ~~5~~57. Hardware beamformed TFM with imasonic 128 elements 15MHz center frequency, 0.25 pitch probe, 36x50mm, 256 x 750pixels.

4. CONCLUSION

This work presents a high-end flexible and scalable hardware architecture for ultrasound imaging systems, able to operate in real-time by hardware beamforming, or transferring the raw data to a high-end processing computer for software beamforming. A 128 channel was built and tested in three demanding applications: Pre-clinical functional ultrasound brain imaging, 3D autofocused imaging in non-destructive testing, and real-time inspection with TFM technique. Future work includes extending the system channel count in a new prototype, able to work with 1024 elements matrix arrays for 3D functional ultrasound.

5. ACKNOWLEDGMENTS

Supported by the projects PID2022-143271OB-I00, funded by MCIN/AEI /10.13039/501100011033/FEDER, UE, PLEC2024-011165, funded by MICIU/AEI/10.13039/501100011033/ FEDER, UE, and by TEC-2024/TEC-43, LUNABRAIN-CM, funded by Comunidad de Madrid.

6. REFERENCES

[1] "*AMBA® AXI™ and ACE™ Protocol Specification AXI3™, AXI4™, and AXI4-Lite™ ACE and ACE-Lite™*", Arm Holdings plc, Cambridge, United Kingdom

[2] "*SP011Aurora 64B/66B Protocol Specification*", Advanced Micro Devices, Inc., Santa Clara, California, United States of America

[3] J. F. Cruza, "Design a real time beamformer", *Trabajo Fin de Maestría*, Dpto. de Electrónica, Esc. Politécnica Superior, Univ. Alcalá de Henares, Sep. 2011.

[4] T. I. Laakso, V. Valimaki, M. Karjalainen, U.K.Laine, "Splitting the Unit Delay", *IEEE Sig. Proc. Magazine*, pp. 30-58, Jan. 1996.

[5] G. S. Kino, D. Corl, S. Bennett, K. Peterson, "Real time synthetic aperture imaging system", *Proc. IEEE Ultrason. Symp.*, pp. 722-731, 1980.

[6] "*UG479- 7 Series DSP48E1 Slice*", Advanced Micro Devices, Inc., Santa Clara, California, United States of America

[7] M. Njiki, S. Bouaziz, A. Elouardi, O. Casula, O. Roy, "A Multi-FPGA implementation of real-time reconstruction using total focusing method", *Proc. 2013 IEEE Int'l Conf. on Cyber Technology in Automation, Control and Intelligent systems* , pp. 468-473, May 26-29, Nanjing, China, 2013.

[8] J. F. Cruza, J. Camacho, L. Serrano, C. Fritsch, "New Method for Real-time Focusing through Interfaces", *IEEE Trans. on Ultrason. Ferroelectr. Freq. Control*, **60,** 4, pp. 739-751, 2013.

[9] C. Fritsch, J. F. Cruza, J. Camacho, J. M. Moreno, J. Brizuela, L. Medina, "Controlador de enfoque dinámico para sistemas de imagen ultrasónica", *Pat. ES P201230799*, 25 May 2012.

[10] G. Montaldo, M. Tanter, J. Bercoff, N. Benech, and M. Fink, "Coherent plane-wave compounding for very high frame rate ultrasonography and transient elastography," *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, 56, 3, pp. 489–506, 2009.

[11] G. Cosarinsky, J.F. Cruza, M. Muñoz, J. Camacho, "Optimized auto-focusing method for 3D ultrasound imaging in NDT", *NDT & E International*, Volume 134, 2023, 102779, ISSN 0963-8695, https://doi.org/10.1016/j.ndteint.2022.102779.