Technical University of Denmark

DTU

An Architecture and Implementation of Real-time Synthetic Aperture Compounding with SARUS

Stuart, Matthias Bo; Jensen, Jørgen Arendt

Link back to DTU Orbit

DTU Library
Technical Information Center of Denmark

Paper presented at the IEEE International Ultrasonics Symposium, Orlando Florida, 2011:

An Architecture and Implementation of Real-time Synthetic Aperture Compounding with SARUS

Matthias Bo Stuart and Jørgen Arendt Jensen

Center for Fast Ultrasound Imaging,

Biomedical Engineering group, Department of Electrical Engineering, Bldg. 349,
Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark

An Architecture and Implementation of Real-time Synthetic Aperture Compounding with SARUS

Matthias Bo Stuart and Jørgen Arendt Jensen
Center for Fast Ultrasound Imaging, Dept. of Elec. Eng. Bldg. 349,
Technical University of Denmark, DK-2800 Lyngby, Denmark

Abstract—**Synthetic aperture and compounding are imaging techniques for increasing the resolution and contrast of ultrasound images. Both techniques are computationally intensive, and combined they require approximately two orders of magnitude more lines to be beamformed per second compared to conventional B-mode imaging with similar frame rates. In this paper, an implementation of a system capable of synthetic aperture compound imaging in real-time producing more than 325 million complex beamformed samples per second is presented. This corresponds to synthetic aperture compound imaging at 13 frames per second with 64 emissions and 3 compound angles with 128 lines each. The beamformer is implemented in the SARUS research scanner which consists of 320 Virtex4 FPGAs and has 1024 independent transmit and receive channels. The beamformer is partitioned across 64 FPGAs and runs at 87.5 MHz while consuming 76% of the available logic resources in each FPGA. The beamformed images have resolution similar to offline processed images.**

I. Introduction

Synthetic aperture (SA) imaging [1], [2] makes it possible to synthesize dynamic transmit and receive focusing in many directions simultaneously. In transmit, a set of transducer elements is emitting the same waveform with individual delays to create a virtual transmit point source. On each emission, all image lines are beamformed, using different weights per line and per virtual source, creating a low-resolution image. The final high-resolution image is created by adding the low-resolution images for all the virtual sources. The process is illustrated in Fig. fig:sa. The advantages include higher resolution and potentially faster acquisition times, as fewer emissions are required compared to conventional focused ultrasound emissions. The main drawback is that the computational complexity increases by two orders of magnitude, as each emission results in a full image rather than a single or a few lines.

Compounding can be done in both transmit [3] and in receive [4]. In transmit, a region is insonified from multiple angles by electronically steering the beam. In receive, a region is imaged from multiple angles by beamforming lines in different directions. Compounding improves speckle-to-noise ratio at a cost of increased computational complexity, as the number of lines to beamform grows linearly with the number of compounding angles.

Synthetic aperture compounding (SAC) is the mixture of SA and compounding, where a full set of image lines for each compounding angle is beamformed for each virtual source. Compared to conventional compounding, where the region of interest needs to be insonified at different beam angles, SAC does not suffer from frame-rate reduction, as each emission of ultrasound energy results in image lines for each compound angle. Furthermore, SAC has the advantage of SA imaging allowing dynamic focusing in every point in the image.

Assuming a setup with 64 virtual sources, i.e., 64 emissions to create a high-resolution image, 3 compounding angles with 128 image lines each, and 13 frames per second, $319,488$ lines need to be beamformed per second. A conventional compound imaging system with one line beamformed per emission, 128 image lines, 3 angles, and 13 compounded frames per second, needs $4,992$ lines to be beamformed per second. The computational complexity is, thus, roughly two orders of magnitude greater than conventional compound imaging.

This contribution of this paper is an architecture and implementation of real-time SAC on the synthetic aperture real-time ultrasound system (SARUS) research scanner.

The rest of the paper is structured as follows: In Section II, SARUS is presented. Section III presents the architecture of the beamformer and compounding, and Section IV describes the conducted experiments to evaluate the beamformer. Section V presents the results, while Section VI concludes the paper.

II. SARUS

SARUS is a research scanner constructed at the Technical University of Denmark [5], [6]. The scanner consists of an analog front-end with 1024 independent transmit and receive channels and a digital back-end. The back-end consists of 64 digital acquisition and ultrasound processing (DAUP) boards. Each DAUP board has 16 independent transmit and receive channels that are sampled at 12 bits and 70 MHz.

Fig. 2 shows a schematic of a DAUP board. Each DAUP board hosts five field-programmable gate arrays (FPGAs) for control, transmission, filtering, focusing, and summing of ultrasound data. The FPGAs all have external memory for storage of parameters and ultrasound data. The control FPGA is a Xilinx Virtex4 FX60, while the remaining four FPGAs are Xilinx Virtex4 FX100. These FPGAs contain $42,176$ slices of logic, various input/output elements, and 160 high-speed DSP slices capable of running at 400 MHz giving 64 billion operations per second maximum multiplication rate per FPGA.

The receive FPGA on each board receives sampled data from the ADCs for 16 channels. From these, sampled data for all 16 channels can be stored in RAM, while 4 channels can be selected for real-time processing. The selection of channels

for processing is fully flexible and can be adjusted for each emission allowing the use of a sliding aperture.

The samples from the selected channels are passed through a programmable filter in the frequency domain and the result is then sent to the focusing FPGA. Here, the samples are beamformed using parametrized focusing units that produce high-resolution tiles (a tile is the contribution to a set of lines from the receive channels on a given DAUP board [7]). These tiles are then sent to the sum FPGA, where they are summed with tiles from other DAUP boards to form the final high-resolution image.

III. IMPLEMENTATION

The implementation is based on a delay-and-sum beamformer, but partitioned to minimize the amount of data that is transferred between DAUP boards [7]. Given the coordinates of the ith virtual source, $\overrightarrow{p}_{vs,i}$, the coordinates of the jth transducer element, $\overrightarrow{p}_{rx,j}$, and the kth point to beamform on the lth line in the output image, $\overrightarrow{p}_{bf,k,l}$, the time of flight is calculated as

$$t_{i,j,k,l} = \frac{\left|\overrightarrow{p}_{bf,k,l} - \overrightarrow{p}_{vs,i}\right| + \left|\overrightarrow{p}_{bf,k,l} - \overrightarrow{p}_{rx,j}\right|}{c} \quad (1)$$

where c is the speed of sound.

Given a sampled and filtered signal $S_{i,j}$ for the ith virtual source on channel j, the focusing index m corresponding to the calculated time of flight is

$$m_{i,j,k,l} = t_{i,j,k,l} f_s \quad (2)$$

where f_s is the sampling frequency.

The output sample $\mathcal{S}_{k,l}$ of a synthetic aperture beamformer with N input channels and M virtual sources is then the sum

of the individually delayed signals

$$\mathcal{S}_{k,l} = \sum_{i=1}^{M} \sum_{j=1}^{N} S_{i,j}(m_{i,j,k,l}) \quad (3)$$

The computation of $\mathcal{S}_{k,l}$ can be partitioned in tiles that match the number of channels on a DAUP board. Let N_{DAUP} be the number of DAUP cards in the system, N_{local} is the number of channels on a DAUP card, and j_d is the jth local channel on DAUP card d. (3) can then be rewritten as

$$\mathcal{S}_{k,l} = \sum_{d=1}^{N_{\text{DAUP}}} \sum_{i=1}^{M} \sum_{j=1}^{N_{\text{local}}} S_{i,j_d}(m_{i,j_d,k,l}) \quad (4)$$

Apodizations (windowing) are typically used in the beamforming process to lower side-lobes at a cost in resolution. Let $w_{vs,i,j,k,l}$ be the apodization of virtual source i for channel j, output sample k on line l, and $w_{rx,j,k,l}$ be the apodization of channel j, output sample k on line l. Adding apodizations to the expression for the beamformed sample in (4) gives

$$\mathcal{S}_{k,l} = \sum_{d=1}^{N_{\text{DAUP}}} \sum_{i=1}^{M} \sum_{j=1}^{N_{\text{local}}} S_{i,j_d}(m_{i,j_d,k,l}) w_{vs,i,j_d,k,l} w_{rx,j_d,k,l} \quad (5)$$

The apodization curves are procedurally generated as a function of the output sample index k from a description of eight line segments each having a start value, a slope, and a length.

The formulation of the synthetic aperture beamformer in (5) allows the inner two sums to be calculated locally on each DAUP board, minimizing the amount of inter-board communication.

The implementation of the inner sums of (5) is illustrated in Fig. 3. The beamformer processes four channels per DAUP board per virtual source. The updating of virtual source and transducer coordinates is not shown in the illustration. Five
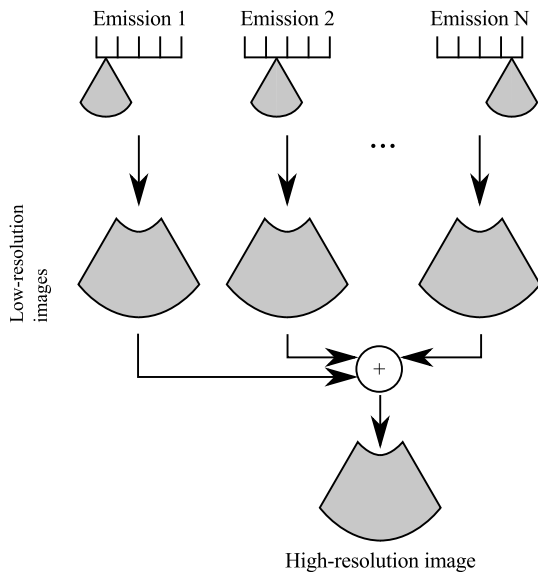

Fig. 1. Synthetic aperture imaging: Each emission results in a low-resolution image all of which are added to form the final high-resolution image.
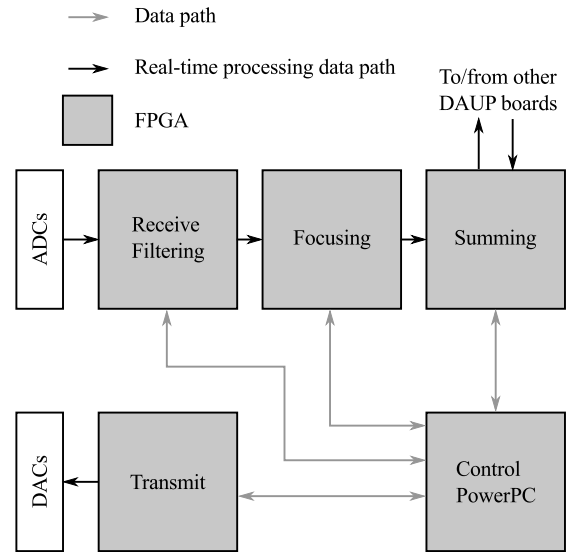

Fig. 2. A schematic of the DAUP board illustrating the major components and data paths in the system. The non-real-time data paths are used for configuration and reading out sampled and processed data.
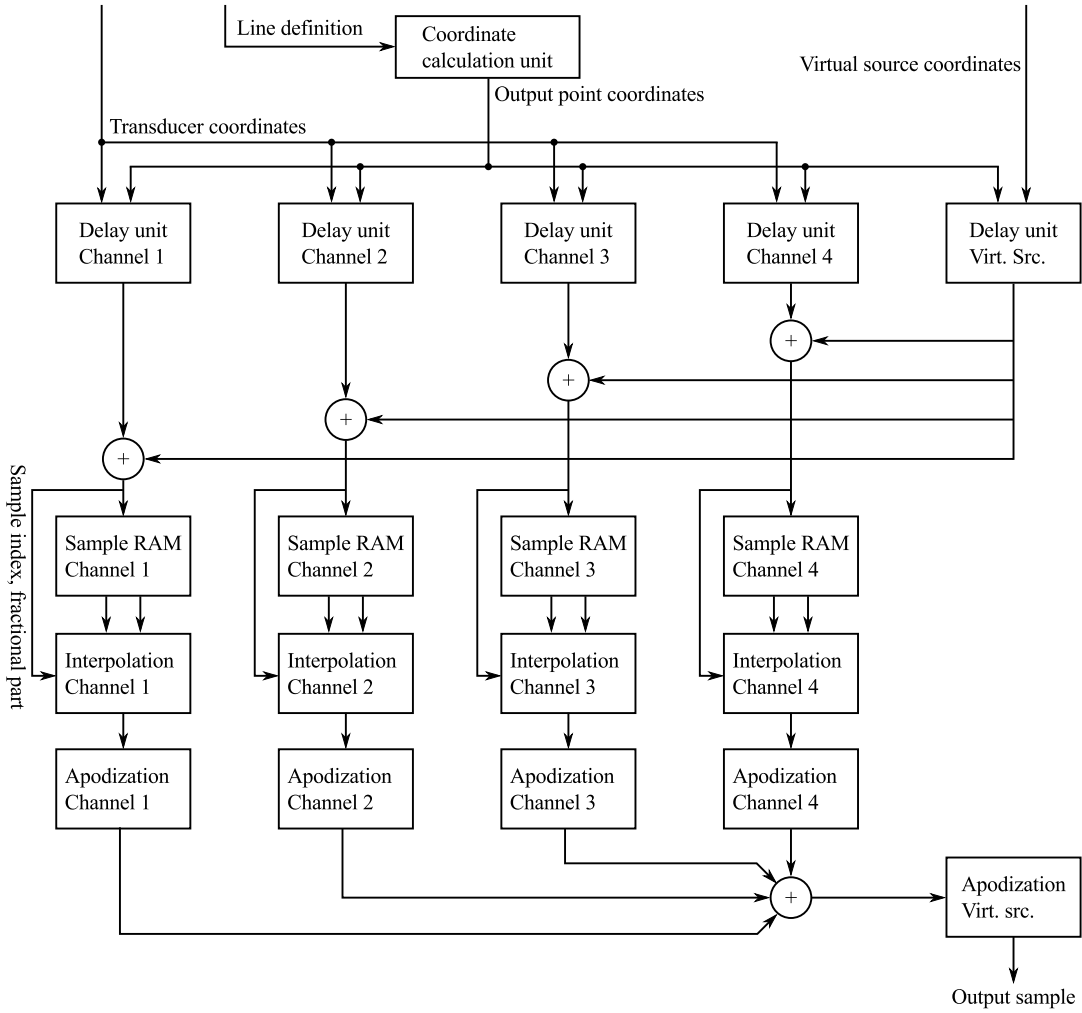
Fig. 3. The architecture of the beamformer.

delay units calculate the four focusing indexes, m. One calculates the distance from the virtual source to the point being beamformed, while the other four calculate the distances from the four transducer elements to the point being beamformed. The integral part of the focusing index is used to index in the sample memory, while the fractional part is used to perform a linear interpolation between two neighbouring samples.

Each $S_{i,j_d}(m)$ is apodized with the receive apodization $w_{rx,j_d,k,l}$, the apodized signals are summed across local channels (j_d) and then apodized with the virtual source apodization $w_{vs,i,j,k,l}$. The beamformer sequentially produces each output sample k on line l for virtual source i and stores the intermediate value

$$S'_{i,k,l} = \sum_{1}^{i} \sum_{j=1}^{N_{\text{local}}} S_{i,j_d}(m_{i,j_d,k,l}) w_{vs,i,j_d,k,l} w_{rx,j_d,k,l} \quad (6)$$

where $S_{k,l} = S'_{M,k,l}$.

The beamformer is pipelined and has a throughput close to one output sample per clock cycle – it takes as many clock cycles as there are pipeline stages to fill the pipeline. When the pipeline is full, one pixel is output per clock cycle. It takes

1,075 clock cycles to beamform a line with 1,024 complex output samples.

Due to constraints on the number of computational and memory resources in an FPGA, the beamformer produces tiles of eight lines at a time. This gives a good trade-off between the utilization of on-chip resources and bandwidth usage to external memory. Multiple passes are made over the beamforming algorithm to produce multiple tiles.

Compounding is performed by beamforming multiple sets of lines at different angular offsets. Fig. 4 illustrates the principle in SARUS, when the beamformer is programmed with 384 lines. Lines 1 to 128 are beamformed at an angle of +30 degrees, lines 129 to 256 at an angle of 0 degrees, and lines 257 to 384 at an angle of -30 degrees.

Each sub-image consisting of all lines at a given angle is sent to a terminal over a 1 Gbit network connection. The sub-images are then envelope detected and scanline converted. The scanline converted sub-images are added to form the final high-resolution compounded image, which is logarithmically compressed for display.
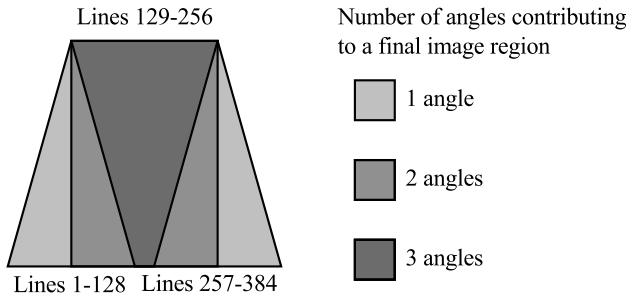
Fig. 4. 384 lines are beamformed. Lines 1 to 128 at an angle of +30 degrees, lines 129 to 256 at 0 degrees, and lines 257 to 384 at -30 degrees.

IV. Experimental Evaluation

The architecture described in the previous section is described in VHDL with supporting peripherals (such as memory controllers and multi gigabit transceivers) generated using Xilinx Core Generator. The code is then implemented in a Xilinx xc4vfx100-10ff1152 FPGA.

Simulated ultrasound data generated with Field II [8], [9] is then beamformed on a system consisting of four DAUP boards. A reference image from the same data is made with the BFT3 beamformation toolbox [10]. The simulation uses a 64 element $\lambda/2$ pitch linear array transducer with a center frequency of 3.5 MHz and an elevation focus at 7 cm. The simulated phantom consists of 3 wires at depths of 30, 35, and 40 mm and centered relative to the transducer.
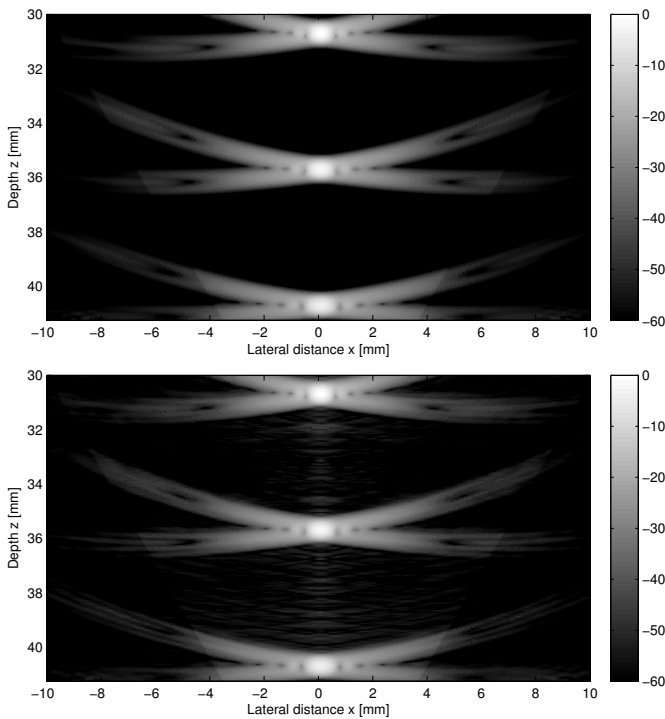

Fig. 5. Top: Reference image beamformed using BFT3. Bottom: Image beamformed with SARUS.

V. Results

The fully implemented beamformer with all peripherals occupies 76% of the logic slices, 64% of the block RAMs, and 60% of the XtremeDSP slices in this FPGA.

The design has been implemented at a core clock frequency of 87.5 MHz. At this frequency, it takes approximately 25 μs to beamform a set of eight lines for a single virtual source. In a setup with 64 virtual sources and 384 lines (48 sets of 8 lines), it thus takes 77 ms to beamform a full image giving a frame-rate of 13 frames per second or 320,000 lines per second or 325 million beamformed samples per second per channel. For a full system beamforming 256 channels per emission, the total output is 83,2 billion beamformed samples per second.

Fig. 5 shows the reference image and the image beamformed with SARUS. Both images show similar resolution (FWHM) of 1.2λ axially, and 1.5λ, 1.7λ, and 1.9λ laterally for the three wires respectively.

VI. Conclusion

This paper has presented an architecture and implementation for making real-time synthetic aperture compound images. The beamformer is implemented at a clock frequency of 87.5 MHz and takes up 76% of the logic resources in a Xilinx Virtex4 FPGA. At this clock frequency, it produces 320,000 beamformed lines per second.

References

[1] M. Soumekh, *Synthetic aperture radar. Signal processing with MATLAB algorithms*. New York: John Wiley & Sons, Inc., 1999.

[2] J. J. Flaherty, K. R. Erikson, and V. M. Lund, "Synthetic aperture ultrasound imaging systems," United States Patent, US 3,548,642, 1967, Published 22 Dec 1970.

[3] D. Shattuck and O. von Ramm, "Compounding scanning with a phased array," *Ultrason. Imaging*, vol. 4, pp. 93–107, 1982.

[4] S. K. Jespersen, J. E. Wilhjelm, and H. Sillesen, "Multi-angle compound imaging," *Ultrason. Imaging*, vol. 20, pp. 81–102, 1998.

[5] J. A. Jensen, M. Hansen, B. G. Tomov, S. I. Nikolov, and H. Holten-Lund, "System architecture of an experimental synthetic aperture real time ultrasound system," in *Proc. IEEE Ultrason. Symp.*, Oct. 2007, pp. 636–640.

[6] J. A. Jensen, H. Holten-Lund, R. T. Nielson, B. G. Tomov, M. B. Stuart, S. I. Nikolov, M. Hansen, and U. D. Larsen, "Performance of SARUS: A Synthetic Aperture Real-time Ultrasound System," in *Proc. IEEE Ultrason. Symp.*, Oct. 2010, pp. 305–309.

[7] S. Nikolov, J. A. Jensen, and B. G. Tomov, "Fast parametric beamformer for synthetic aperture imaging," *IEEE Trans. Ultrason., Ferroelec., Freq. Contr.*, vol. 55, no. 8, pp. 1755–1767, 2008.

[8] J. A. Jensen, "Field: A Program for Simulating Ultrasound Systems," *Med. Biol. Eng. Comp.*, vol. 10th Nordic-Baltic Conference on Biomedical Imaging, Vol. 4, Supplement 1, Part 1, pp. 351–353, 1996.

[9] J. A. Jensen and N. B. Svendsen, "Calculation of Pressure Fields from Arbitrarily Shaped, Apodized, and Excited Ultrasound Transducers," *IEEE Trans. Ultrason., Ferroelec, Freq. Contr.*, vol. 39, pp. 262–267, 1992.

[10] J. M. Hansen, M. C. Hemmsen, and J. A. Jensen, "An object-oriented multi-threaded software beam formation toolbox," in *Proc. SPIE - Medical Imaging - Ultrasonic Imaging and Signal Processsing*, vol. 7968, 2011, p. 79680Y.