Parametric Beamformer for Synthetic Aperture Ultrasound Imaging

Nikolov, Svetoslav; Tomov, Borislav Gueorguiev; Jensen, Jørgen Arendt

Published in:
IEEE Ultrasonics Symposium

Link to article, DOI:
10.1109/ULTSYM.2006.504

Publication date:
2006

Document Version
Publisher's PDF, also known as Version of record

Citation (APA):
Nikolov, S., Tomov, B. G., & Jensen, J. A. (2006). Parametric Beamformer for Synthetic Aperture Ultrasound Imaging. In IEEE Ultrasonics Symposium (pp. 2172-2176). IEEE. DOI: 10.1109/ULTSYM.2006.504

Parametric Beamformer for Synthetic Aperture Ultrasound Imaging

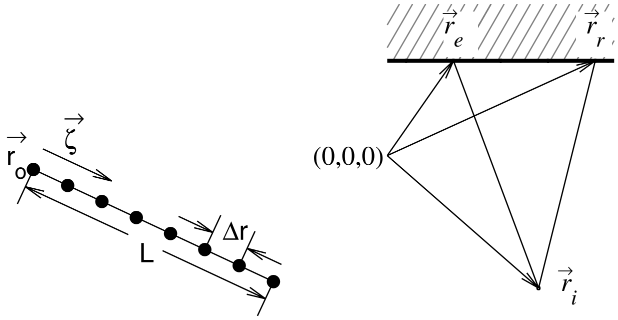
Svetoslav I. Nikolov, Borislav G. Tomov, Jørgen A. Jensen
Center for Fast Ultrasound Imaging, Bldg 348, Ørsted•DTU, Technical University of Denmark
DK-2800 Lyngby, Denmark

Abstract— **In this paper a parametric beamformer, which can handle all imaging modalities including synthetic aperture imaging, is presented. The image lines and apodization coefficients are specified parametrically, and the lines can have arbitrary orientation and starting point in 3D coordinates. The beamformer consists of a number of identical beamforming blocks, each processing data from several channels and producing part of the image. A number of these blocks can be accommodated in a modern field-programmable gate array device (FPGA), and a whole synthetic aperture system can be implemented using several FPGAs. For the current implementation, the input data is sampled at 4 times the center frequency of the excitation pulse and is match-filtered in the frequency domain. In-phase and quadrature data are beamformed with a sub-sample precision of the focusing delays of 1/16th of the sampling period. Each line is completely specified by 3 input parameters. The focusing delays are calculated iteratively in a 8-stage deep pipeline, and focusing information for 8 different lines is interleaved to produce delays at every clock cycle. The apodization is specified using piecewise linear approximation with 255 levels. A beamforming block uses input data from 4 elements and produces a set of 10 lines. Linear interpolation is used to implement sub-sample delays. The VHDL code for the beamformer has been synthesized for a Xilinx V4FX100 speed grade 11 FPGA, where it can operate at a maximum clock frequency of 167.8 MHz. Each beamformation block requires 12 multipliers, 5 buffers for parameters, 8 buffers for input data and 32 buffers for output data (I and Q). Furthermore double-buffering is used for the input data, thus simplifying the synchronization. Up to six beamforming blocks can fit in one FPGA. Clocked at 150 MHz they produce 900×10^6 I and Q samples/second. Assuming a pulse repetition frequency of 5000 Hz, these blocks can be configured to beamform in real time 256 B-mode lines of synthetic aperture data from 4 transducer elements, or 64 lines from 16 elements.**

I. Introduction

Medical ultrasound imaging is a widely used imaging modality, which is characterized by high mobility and short preparation time. The current sequential line acquisition image generation approach in commercial scanners has been around for 30 years and is essentially based on sequential processing [13]. A very promising algorithm for ultrasound image generation is synthetic aperture (SA) imaging [1], [3], [9], [10], [12], which provides uniform resolution across the image and fast data acquisition. The latter is especially desired for cardiac examinations. The SA technique also makes feasible 3D imaging, in which a large number of image lines/points/planes has to be created quickly for the purpose of real-time display.

Storing and accessing the focusing information for each image line is a major problem for the real-time beamformers implementing synthetic aperture imaging or advanced flow imaging techniques. The focusing precision is of paramount importance for the image quality [4] and simple compression is not an option. Parametric recursive delay generators have been suggested for the case of image lines originating from the transducer [2], [14]. For the purpose of the vector flow imaging [5]–[8], though, a free choice of origin and direction of the lines have to be possible.



(a) (b)

Fig. 1. Line definition (a) and propagation path (b)

The purpose of this paper is to present a parametric beamformer capable of fast focusing in an arbitrary direction in 3D space, using a recursive parametric delay generation algorithm, requiring only 3 input parameters per line [11]. The algorithm uses successive approximation with error accumulation/compensation. In the current hardware implementation, the delay approximation is pipelined, and the parameter sets for 8 lines are fed into the pipeline in an interleaved fashion to keep all pipeline stages active.

Section II presents the theory behing the beamformer, Section III describes the implementation of the beamformer in hardware, and Section IV presents the performance estimates and the resource utilization with a Xilinx V4FX100 (speed grade 11) FPGA.

II. Beamformer overview

The presented beamformer handles images acquired using the conventional line-by-line acquisition as well as synthetic aperture techniques. In conventional imaging a focused wave is transmitted from the transducer in a given direction. Echoes are scattered back by inhomogeneities on the path of the propagating wave. The received signals are coherently summed to form a beam. To reconstruct the reflectivity at given spatial location, the distance from the origin of the beam to that location and back to the transducer element that has recorded the echo signal must be calculated.

A spherical wave rather than a focused wave is transmitted to acquire data in synthetic *transmit* aperture imaging. In receive all element are used to record the back-scattered signal. A full image is reconstructed for every emission. It has low resolution, because there is no transmit focusing. The measurement is repeated by transmitting a spherical wave with another transducer element and a new low resolution image is created. After all elements have been used in transmit, all low resolution images are summed and an image with high-resolution is formed.

Delay-and-sum beamforming is used to reconstruct images for both SA and conventional imaging. Because each transmission covers the whole region of interest, the notion of ”beams” loses its meaning in the case of synthetic aperture imaging, where an image could be specified as a set of picture elements. The distance to the reconstructed points in SA images does not necessarily increase from point to point, and the sampled RF data must be stored until all points have been beamformed. The beamforming of the high resolution image $H(\vec{r}_i)$ at spatial location $\vec{r}_i$ can be expressed as:

$$H(\vec{r}_i) = \sum_{e=1}^{N} a_e(\vec{r}_i) \sum_{r=1}^{M} a_r(\vec{r}_i) s_r(\tau_{ToF}(\vec{r}_i, \vec{r}_e, \vec{r}_r)), \tag{1}$$

where τ_{ToF} is the time of flight of the echo $s_r(t)$ received by transducer element at $\vec{r}_r$, after an emission with transducer element located at $\vec{r}_e$ (see Fig. 1(b)). The coefficients a_e and a_r are transmit and receive apodization, respectively. Digital beamformers operate with sampled signals and interpolation is used to reconstruct the signal $s_r(\tau_{ToF})$ when τ_{ToF} is not an integer multiple of the sampling period. The time of flight t_{ToF} is the sum of forward and backward propagation times τ_f and τ_b, which can be calculated with the same algorithm. In the rest of the paper we will consider only one of the propagation times and the subscript indicating a forward or backward propagation will be omitted.

For convenience the image points are placed along lines defined by an origin $\vec{r}_o$, direction $\vec{\zeta}$, length L and distance between two points Δr as shown in Fig. 1(a). The coordinates $\vec{r}_i = (x, y, z)^T$ of the i:th point along the line are:

$$\vec{r}_i = \vec{r}_o + i\Delta\vec{r}. \tag{2}$$

The distance l_i from the transmission origin $\vec{r}_e$ to a point $\vec{r}_i$ is:

$$l_i = |\vec{r}_o - \vec{r}_e + i\Delta\vec{r}| = |\vec{r}_{oe} + i\Delta r|, \tag{3}$$

where $\vec{r}_{oe}$ is the origin of the line expressed in coordinates relative to the position $\vec{r}_e$ of the element, the distance to which is sought.

For each of the three coordinates the squared distance can be expressed as

$$(x_{oe} + i\Delta x)^2 = x_{oe}^2 + 2ix_{oe}\Delta x + (i\Delta x)^2. \tag{4}$$

The value at the previous focal point is

$$x_{oe}^2 + 2(i-1)x_{oe}\Delta x + ((i-1)\Delta x)^2. \tag{5}$$

Subtracting (5) from (4) results in

$$l_i^2 - l_{i-1}^2 = 2x_{oe}\Delta x + \Delta x^2(2i-1), \tag{6}$$

which is the increment from sample to sample, when performing the focusing. The origin of the line x_{oe} and the increment from sample to sample Δx are constants and can be precalculated.

The difference between the distances to two consecutive points squared $\Lambda_i = L_i - L_{i-1} = l_i^2 - l_{i-1}^2$ is:

$$\Lambda_i = A + (2i-1)B, \tag{7}$$

where the constants A and B are calculated as:

$$\begin{aligned} A &= 2(x_{oe}\Delta x + y_{oe}\Delta y + z_{oe}\Delta z) \\ B &= \Delta x^2 + \Delta y^2 + \Delta z^2. \end{aligned} \tag{8}$$

x_{oe}, y_{oe}, z_{oe} are the components of the vector $\vec{r}_{oe}$ from (3). The square of the distance $L_i = l^2$ can be recursively found from the previous squared distance: $L_i = L_{i-1} + \Lambda_i$. Multiplying this expression with $(f_s/c)^2$ gives the squared sample index corresponding to the propagation time. It has been shown in [11] that only Λ_i is needed to recursively calculate $\tau_i = \frac{f_s}{c} l_i$.

The apodization coefficients are also parametrically calculated using piece-wise linear approximation of the ideal curve.

III. Implementation

A block diagram of the developed unit is shown in Fig. 2. An image is specified as a set of lines to accommodate both SA and conventional imaging, and each unit can beamform 8 lines in parallel using data from 4 channels. A beamformer unit contains 5 delay calculation units - 4 for the path to the four receive elements and 1 for the transmit path. The apodization coefficients and delays are calculated parametrically. Linear interpolation is used to generate samples with sub-sample delay precision. The sum of the 4 receive channels represents a partial low-resolution image in the case of SA imaging. It is multiplied with the coefficient for transmit apodization before it is summed with the rest of the low-resolution images.

A. Delay generation

The delay-generation unit recursively calculates the propagation time to points along an image line (see Fig. 1(a)). It consists of two blocks as shown in Fig. 3(a). The first block calculates the difference $\Lambda_i = \tau_{i-1}^2 - \tau_i^2$, where τ_i is the propagation time from the focal point i to a transducer element. Using this difference and the time of propagation from a transducer element to the origin of the image line, it is possible to recursively calculate the distance to all points along the line using the RISQRT unit shown in Fig. 3(c).

Figure 3(b) shows the block diagram of the unit calculating Λ. It consists of three registers B, M and C. M holds the product $2iB$ and C is a constant $C = A - B$. The values of A and B are calculated according to (8) and scaled with f_s/c.

The difference Λ is further sent to the RISQRT (Recursive Iterative SQuare RooT) circuit. It calculates an approximate propagation time $\tau_i \approx \sqrt{T_i}$. The operation of the circuit is based on the fact that $|\tau_i - \tau_{i-1}| \leq \Delta t$ [11], where $\Delta t = \Delta r \frac{f_s}{c}$. The

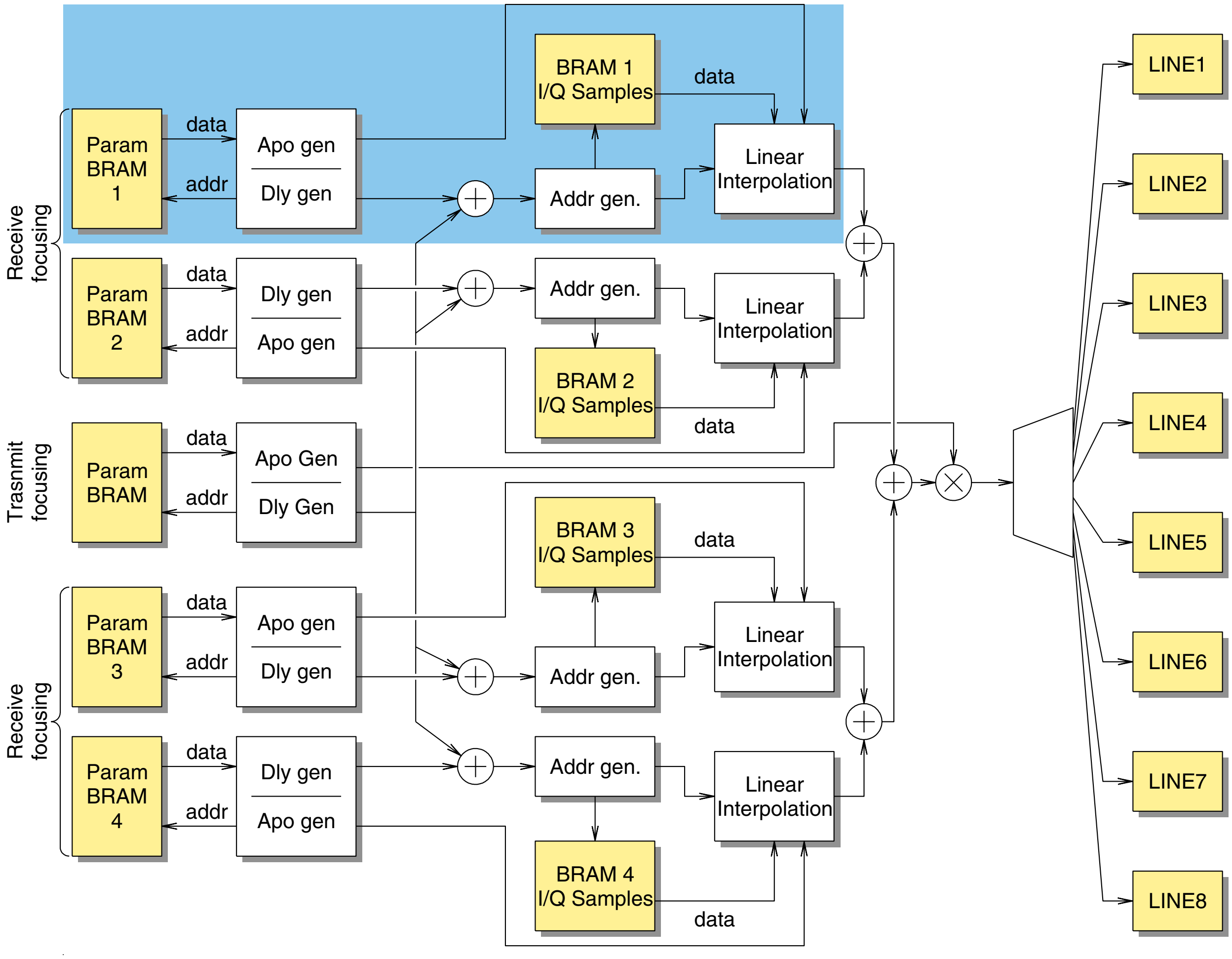
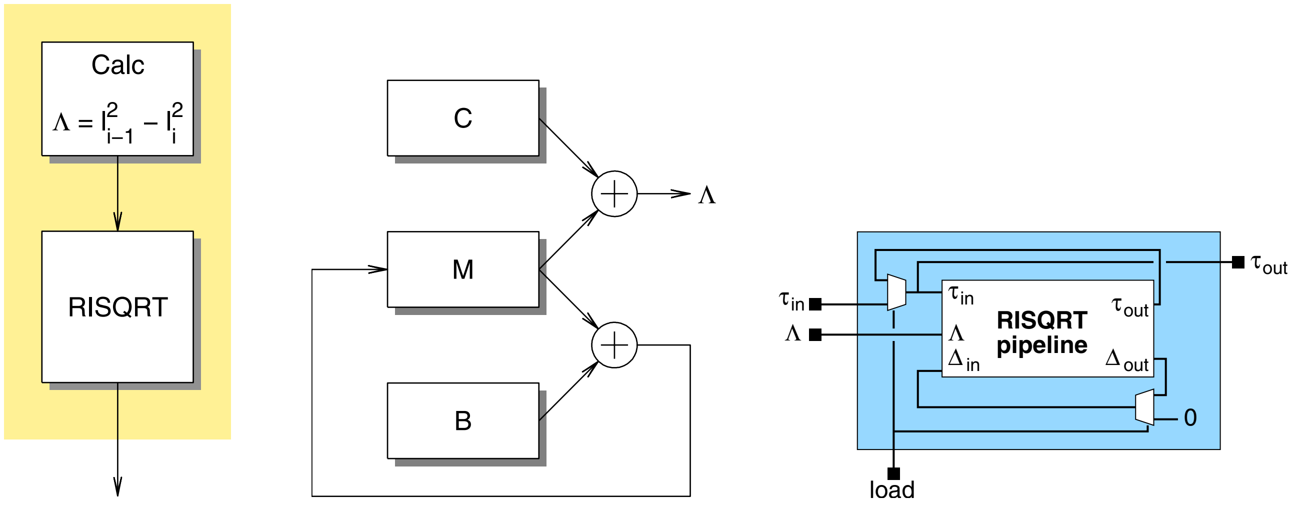


Fig. 2. A single beamforming block produces 8 lines in parallel using data from 4 input channels.



(a) Structure of delay generator

(b) Difference between distances squared

(c) Calculate propagation time

Fig. 3. Structure of the parametric delay generation unit.

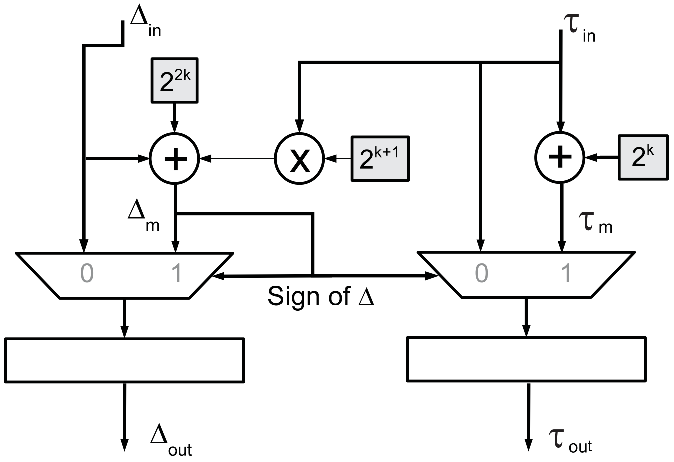


Fig. 4. A single stage of the RISQRT pipeline.

search for the right value of τ_i starts from τ_{i-1}. To decrease the complexity of the circuit, the search starts always from a value that is less than τ_i. In other words, if $\Lambda_i < 0$ then the start value for the search is $\tau_{start} = \tau_{i-1}$, otherwise it is $\tau_{start} = \tau_{i-1} - \Delta t$. The circuit consists of a pipeline which is 8 stages deep.

Each stage adds one bit of precision to the estimate. The output of the pipeline is the estimate of the propagation time τ_i and the residual error Δ_i, where $\Delta_i = \tau_i^2 - \sqrt{T_i}$. Both the estimate τ_i and the residual error Δ_i are fed back to the input as shown in Fig. 3(c). The k:th stage (shown in Fig. 4) performs the following operation:

$$\tau_m = \tau_{in} + \varepsilon(k) \tag{9}$$

$$\tau_{out} = \begin{cases} \tau_m, & \tau_m^2 \leq T_i \\ \tau_{in} & \text{otherwise}, \end{cases} \tag{10}$$

where $\varepsilon(k)$ is the step size at k:th iteration. In the start ε is equal to Δt. At every next stage, the step is divided by two. τ_{in} and Δ_{in} are the approximation of the propagation time and the residual error calculated in the previous stage. τ_m and Δ_m are the new candidates for the final result. The result Δ_{out} and τ_{out} from the stage depend on the sign of Δ_m. If $\Delta_m < 0$, then $\tau_m < \sqrt{T_i}$ and $\tau_{out} = \tau_m$, else $\tau_{out} = \tau_{in}$.

The initial step size is chosen to be a power of two, and all multiplications are reduced to shifting operations as shown in Fig. 4. Furthermore, each stage is fixed to a given iteration number, and the shifting operation is therefore reduced to a suitable signal wiring.

The pipeline consists of 8 stages, and it is therefore necessary to calculate delays for 8 lines simultaneously to keep the pipeline full. The algorithm needs the delay for the first point in the line to start the recursive procedure. This delay is sent to the output of the RISQRT circuit to avoid idle clock cycles in the result.

B. Apodization

The apodization curves are described using piecewise linear approximation with maximum error of 1% of the full scale. A greedy algorithm is used to find the segments off-line. Each segment is specified by a start value, slope and a segment length in samples. The start value and the slope for each segment are encoded using 14 bits and the number of samples is encoded using 8 bits. Although the internal calculations are performed with 14-bit precision, the result is 8-bit, thus introducing up to 0.4% numerical error.

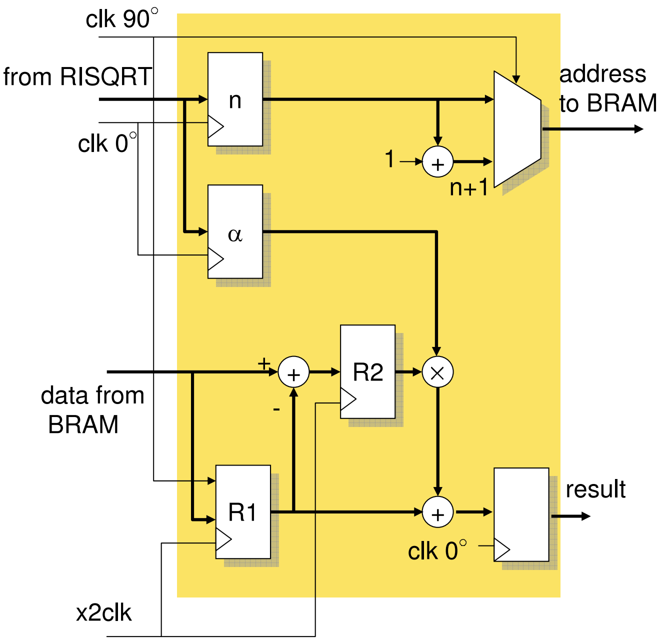


Fig. 5. Structure of the interpolation circuit.

C. Interpolation

Linear interpolation is used to generate samples for time instances that are not integer multiple of the sampling period. The linear interpolation uses the built-in digital signal processing blocks available in the Xilinx Virtex-4 family of FPGAs. Each such block can multiply two 18-bit numbers and accumulate the result in a 48-bit register. The unit must operate at a clock frequency which is twice higher than the clock frequency of the delay generator, because two samples are needed for each output sample. The output sample is calculated as follows:

$$\begin{aligned} \hat{s}[k] &= \alpha s[n+1] + (1-\alpha)s[n] \\ &= s[n] + \alpha(s[n+1] - s[n]) \end{aligned} \tag{11}$$

The index n is the integer part of the delay. It is represented by 12 bits and can address up to 4096 samples. The subsample delay α is formed by the 4 least significant bits of the propagation time. The circuit is shown in Fig. 5.

IV. Resources

Each beamformation unit produces 1 sample at every clock cycle. The circuit is clocked at 150 MHz. The total number of samples is about 1×10^9 (about 200 lines, 1000 samples, 5000 transmissions/second). A total of 6 beamformation blocks are required to beamform SA images in real time. Each block must therefore beamform 4 sets of 8 lines in parallel. The resources necessary for the beamformation fall into 3 categories: built-in block RAM to store parameters, input samples and result; dedicated digital signal processing blocks to perform the interpolation and apodization; and logic resources used to control the circuit and generate delay and apodization coefficients.

A. Block RAM:

The dedicated Block RAM (BRAM) units can be configured to store 512 36-bit words, which is sufficient to hold the parameters for one transducer element for 32 lines. Three 32-bit parameters are required to describe the delays for a single line. Up to eight segments can be used to approximate the apodization curve for each element. The parameters for every segment are packed in 36 bits (14-bit start value and inclination and 8-bit length). The total number of words, thus, is $(32 \times 3 + 8) + (8 \times 4 \times 8) = 360 < 512$, and 1 BRAM is sufficient. The transmit parameters must be changed at every transmission because the transmit element changes from emission to emission, therefore duble buffering is used. The total number of BRAMs for parameters per beamforer unit is thus 6.

In-phase and quadrature signals are beamformed simultaneously. For each channel 4096 16-bit I and Q samples are stored in Block RAM. This requires 64 BRAMs, including those used for double buffering.

Each focusing block produces 32 lines which must be accumulated over all emissions. The precision is set to 24 bits. The number of samples is 1024. Thus, 96 BRAMs are used to store the beamformed data.

A XC4VFX100 FPGA has 376 BRAMs, and can therefore accommodate 2 beamformation units.

B. Dedicated DSP blocks:

Three DSP blocks are used per channel for interpolation and apodization. One extra DSP block provides transmit apodization (see Fig. 2), giving a total of 26 dedicated DSP blocks per beamformation unit to process I and Q data. A XC4VFX100 FPGA has 160 Extreme DSP slices and is not a restricting factor for the implementation.

C. Logic requirements:

Using the synthesis report generated by the Xilinx ISE tool, a beamformation block requires about 3 300 out of 42 176 available slices. The maximum clock frequency for a device of speed grade 11 is 167.8 MHz and is limited by the RISQRT pipeline logic.

V. CONCLUSION

The present paper describes the beamformer building blocks of a synthetic aperture imaging system. The beamformer can beamform groups of 8 lines originating from any point inside the region of investigation and having an arbitrary orientation, inter-sample distance and length. This makes it possible to beamform lines that are suitable for vector flow estimation. Furthermore the calculation of forward and backward propagation times have been separated enabling the calculation of the time flight for conventional and synthetic aperture images. The description of the beamformed lines uses an exact formula in 3 dimensions making it suitable for 3D imaging. The error in the delay calculation is fed back into the delay calculation unit, thus limiting the maximum error to half of the value of the least significant bit. In the presented case, this error is less than $f_s/16$. Adding more stages in the RISQRT unit can increase the precision.

There are two sources of limitations in the presented design. The speed of calculations is limited by the logic involved in the time-of-flight calculations (the RISQRT circuit). Additional pipelining could increase this speed, but in that case the speed limitation will be imposed by the RAM blocks which must deliver 2 input samples for each beamformed pixel.

The second limitation is imposed by available bandwidth, which makes it necessary to use the built-in RAM blocks for buffers. The number of these RAM blocks limit the number of beamformation units per FPGA to 2 if synthetic aperture data are to be beamformed.

VI. ACKNOWLEDGMENTS

This work was supported by grant 26-04-0024 by the Danish Science Foundation and B-K Medical A/S, Herlev, Denmark.

REFERENCES

[1] S. Bennett, D. K. Peterson, D. Corl, and G. S. Kino, “A real-time synthetic aperture digital acoustic imaging system,” in *Acoust. Imaging*, P. Alais and A. F. Metherell, Eds., vol. 10, 1982, pp. 669–692.

[2] H. T. Feldkämper, R. Schwann, V. Gierenz, and T. G. Noll, “Low power delay calculation for digital beamforming in handheld ultrasound systems,” *Proc. IEEE Ultrason. Symp.*, vol. 2, pp. 1763 –1766, 2000.

[3] C. R. Hazard and G. R. Lockwood, “Theoretical assessment of a synthetic aperture beamformer for real-time 3-D imaging,” *IEEE Trans. Ultrason., Ferroelec., Freq. Contr.*, vol. 46, pp. 972–980, 1999.

[4] S. Holm and K. Kristoffersen, “Analysis of worst-case phase quantization sidelobes in focused beamforming,” *IEEE Trans. Ultrason., Ferroelec., Freq. Contr.*, vol. 39, pp. 593–599, 1992.

[5] J. A. Jensen, “A new estimator for vector velocity estimation,” *IEEE Trans. Ultrason., Ferroelec., Freq. Contr.*, vol. 48, no. 4, pp. 886–894, 2001.

[6] ——, “Directional velocity estimation using focusing along the flow direction: I: Theory and simulation,” *IEEE Trans. Ultrason., Ferroelec., Freq. Contr.*, pp. 857–872, 2003.

[7] ——, “Velocity vector estimation in synthetic aperture flow and B-mode imaging,” in *IEEE International Symposium on Biomedical imaging from nano to macro*, 2004, pp. 32–35.

[8] J. A. Jensen and I. R. Lacasa, “Estimation of blood velocity vectors using transverse ultrasound beam focusing and cross-correlation,” in *Proc. IEEE Ultrason. Symp.*, 1999, pp. 1493–1497.

[9] M. Karaman, P. C. Li, and M. O’Donnell, “Synthetic aperture imaging for small scale systems,” *IEEE Trans. Ultrason., Ferroelec., Freq. Contr.*, vol. 42, pp. 429–442, 1995.

[10] S. I. Nikolov, “Synthetic aperture tissue and flow ultrasound imaging,” Ph.D. dissertation, Ørsted•DTU, Technical University of Denmark, 2800, Lyngby, Denmark, 2001.

[11] S. I. Nikolov, J. A. Jensen, and B. G. Tomov, “Recursive delay calculation unit for parametric beamformer,” in *Proc. SPIE - Progress in biomedical optics and imaging*, vol. 6147-13, 2006, pp. 1–12.

[12] M. O’Donnell and L. J. Thomas, “Efficient synthetic aperture imaging from a circular aperture with possible application to catheter-based imaging,” *IEEE Trans. Ultrason., Ferroelec., Freq. Contr.*, vol. 39, pp. 366–380, 1992.

[13] K. E. Thomenius, “Evolution of ultrasound beamformers,” in *Proc. IEEE Ultrason. Symp.*, vol. 2, 1996, pp. 1615–1621.

[14] B. G. Tomov and J. A. Jensen, “Delay generation methods with reduced memory requirements,” in *Proc. SPIE - Med. Imag.*, 2003, pp. 491–500.