Parallel Design for Ultrasound Synthetic Aperture Imaging FPGA

Hui Juan Huang [#*1], Yuan Yu Yu [#*], Jiu Jiang Wang [#*], Wang Meng [#*], Sio Hang Pun [#2], Peng Un Mak [*3], Mang I Vai [#*4]

*Department of Electrical and Computer Engineering, Faculty of Science and Technology, University of Macau, Macau SAR, China

State Key Laboratory of Analog and Mixed-Signal VLSI, University of Macau, Macau SAR, China

[1] huanghuijuan1228@163.com
[2] lodgepun@umac.mo
[4] fstmiv@umac.mo
[3] fstpum@umac.mo

***Abstract*—Synthetic aperture (SA) imaging algorithm, which combines sub-aperture elements to form high resolution image, can improve image quality in vivo ultrasound image. However, large computation resource is generally required for the implementation of SA. In this paper, we propose a parallel structural design for SA imaging algorithm that can be realized in Field Programmable Gate Array (FPGA), so that light weight and miniaturized design can be achieved. The proposed design has been validated by MATLAB and was employed to construct a high resolution ultrasound imaging from the raw data of 128 array transducer elements. The processing time is about 2.6*ms* theoretically, which should be alright for usage.**

Keywords—Ultrasound imaging; FPGA; Parallel design; Beamforming; Synthetic aperture

I. INTRODUCTION

Ultrasound imaging is the most widely-used non-ionized real time diagnostic tool in clinical applications because of its convenience to use and low radiation [1]. Synthetic aperture (SA) imaging has the advantage to improve the image quality than conventional ultrasound imaging [2,3]. Compared with ordinary sequential scanning method applied to medical ultrasound, SA method had been developed for radar systems in the 1950s to reconstruct of high resolution image from successive transmission and reception position [4]. Since the late 1960s and early 1970s it has been investigated and explored to ultrasound imaging [5]. The implementation of the SA method on medical ultrasound imaging has become active since the beginning of 1990s [6,7].

The main focuses of current SA research in ultrasound are to improve the quality of images, to increase the frame rate and to reduce power consumption [3]. SA techniques can be implemented using hardware such as Graphics Processing Unit (GPU) because of its flexibility and high level of parallelism. GPU is suitable for real-time applications but with high power consumption. In recent years, Field Programmable Gate Array (FPGA) has gained popularity for the design based on rapid prototype technology. It can provide an optimized hardware choice suitable for SA algorithm design. By using FPGAs, we can implement ultrasound image data processing using multiple data paths, and pipelining so that high performance, high resource utilization and reduced power consumption system can be achieved. With the availability of the latest design tools, it is a good choice to model FPGA-based designs first in MATLAB to verify the results for ultrasound image by its graphical output.

This research explores the design of SA algorithm for ultrasound imaging with FPGAs. The algorithm was realized in MATLAB and encoded using hardware description language (HDL) for FPGA. The algorithm principle, parallel design and time cost analysis of FPGA are discussed in this article.

II. DESCRIPTION OF SA ALGORITHM

SA method transmits point source firings from different lateral positions element by element at each emission. Each firing's pulse echoes from the medium are acquired over all elements in the receiving aperture. The received data is used to generate one low resolution image (LRI). Subsequently high resolution image (HRI) is acquired through recursive summation of LRIs. The algorithm consists of three main steps as follows.

A. Analytic signal conversion

The analytic signal conversion is the first step in LRI formation. For the received echo of each emission, the signal is first undergone Hilbert transform to obtain the complex components in the analytic signal. Thus, the received echo is transformed into the pixel values for 2D image.

B. Beamforming

In SA imaging, the delayed beamformation for each pixel P_0 requires the calculation of its time-of-flight $\tau_{n,m}(P_0)$, which is determined by the geometric distance from the transmitting source, to the pixel position, and back to the receive element position [2,8]. $d_T(P_0;m)$ and $d_R(P_0;n)$ indicate the transmitting distance for the m^{th} transmitting

978-1-4799-8641-5/15/$31.00©2015 IEEE

element and the receive distance for the n^{th} channel, respectively.

The speed of ultrasound c_0 and $d_T(P_0;m)$, $d_R(P_0;n)$ give the total propagation time $\tau_{n,m}(P_0)$, which is calculated as the following:

$$\tau_{n,m}(P_0) = \frac{d_T(P_0;m)+d_R(P_0;n)}{c_0}. \quad (1)$$

With the total propagation time, we can get the weight λ by (2) and (3) so that the data point $\alpha_{n,m}(P_0)$ between the adjacent analytic data $\alpha_{n,m}(k)$ and $\alpha_{n,m}(k+1)$ can be calculated mathematically by linear interpolation formulated as (4).

$$k = \lfloor f_s \cdot \tau_{n,m}(P_0) \rfloor, \quad (2)$$

$$\lambda = 1 + k - f_s \cdot \tau_{n,m}(P_0), \quad (3)$$

where k is the proximal depth sample number, f_s is the sampling rate of raw data stream, and $\lfloor . \rfloor$ is the floor operator giving the largest integer but less than the operand.

$$\alpha_{n,m}(P_0) = \lambda \cdot \alpha_{n,m}(k) + [1-\lambda] \cdot \alpha_{n,m}(k+1). \quad (4)$$

Then, a line accumulator (5) applies different apodization weights ω_n to the signal received from different element arrays (usually Hanning window), which can suppress the sidelobe effectively and increase contrast in ultrasound imaging while the width of mainlobe is increasing.

$$L_m(P_0) = \sum_{n=1}^{N} \omega_n \cdot \alpha_{n,m}(P_0). \quad (5)$$

C. Recursive compounding

Eventually, all low resolution images obtained from different transmit positions are combined linearly and a high resolution image can be obtained.

$$H_i(P_0) = \sum_{m=i-M+1}^{i} L_m(P_0), \quad (6)$$

where M denotes the number of LRI frame set used to form the HRI, i indicates the i^{th} frame HRI, and m is the index of LRI.

In summary, in the SA algorithm, the Step A is used to convert the receiving elements acquired signal into analytic signal and then, Step B uses the propagating time of the ultrasound pulse together with the analytic signal obtained in Step A to reconstruct the LRIs for each transmitting element. Finally, Step C combines all LRIs into a HRI linearly.

III. ALGORITHM IMPLEMENTATION IN FPGA

It is found that the algorithm of SA imaging can be implemented in parallel structure to save computation time. The prototype of the implementation of SA in FPGA can be found in Fig. 1. In the current design, the FPGA is mainly responsible for parallel processing the SA algorithm and a computer in charge of handling the preliminary non computational intense processing and the auxiliary functions such as I/O data to the FPGA, image storage, and User Interface.

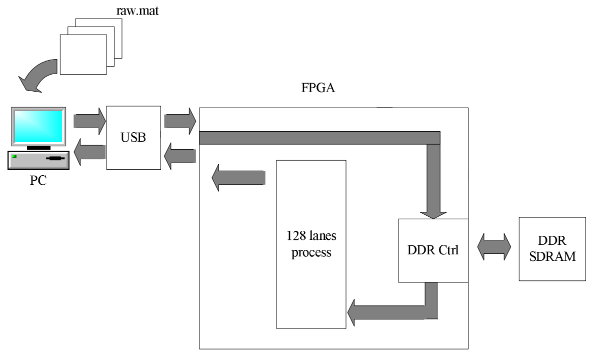

Fig. 1 Design schematic diagram

The proposed method can be applied to process B-mode ultrasound images based on SA data acquisition, which requires full parallel processing in real-time. So we adopt the design of 128 parallel lanes to improve processing capability. The DDR SDRAM is used to store the raw data and the abundant interim processing data.

The detail operation of the proposed design is briefly explained in the following. In our study, the resolution of the acquired ultrasound image is 255×512. The coordinates of the transmitting elements, the receiving elements and reflection point can be expressed respectively as $(T, 0)$, $(R, 0)$ and (x, y), and inputted to the FPGA as y^2, $|T-x|$ and $|R-x|$. The y^2 represents the square of the vertical distance from the transmitter to the reflection point, $|T-x|$ and $|R-x|$, which are denoted as D_t and D_r, indicate the horizontal distances from the transmitting element to reflection point and from the receiver element to reflection point, respectively. For each ultrasound pulse transmission point, T is fixed, so there are 255 values for D_t and 512 for y^2.

To calculate the distance of each pixel to the received elements, since there are R_e received elements working simultaneously, the horizontal distance of each pixel to the each received element should be evaluated. Therefore, the size of D_r is R_e by 255 for each transmitted element.

Now, with D_r and y^2, the distance from each reflection point to the received elements can be calculated and is denoted as R_d, which has a size of 255 by 512. Similarly, by using D_t and y^2, the distance from each reflection point to the transmitted elements can be evaluated as well and is denoted as T_d. By combining T_d and R_d, the propagating distance ultrasound pulse from a particular transmitter reflected by each pixel in the image to form matrix P_d.

P_d represents a reflection distance from the top of the reflection point in the first column to the receive element. This distance determines the starting address when accessing from the raw data stored in memory. Fig. 2 shows two caches shall be prepared for each P_d, each level cache stores 8 bytes of raw data. The start address can be calculated from the first pixel of each frame, the offset can be calculated from the distance between the adjacent pixels. Based on the start address and the offsets, the raw data for all the pixels can be read from the cache and feed into successive steps. These two caches work in ping-pong buffering scheme as illustrated in the figure.

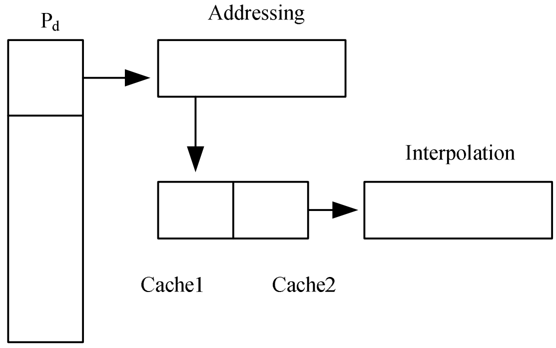

Fig. 2 The processing flow of P_d

Based on the aforementioned procedure to calculate the propagating distance P_d, a parallel design suitable for FPGA implementation is given in Fig. 3. The design is arranged in pipelined architecture so that all R_e columns of data were performed to calculate the propagating distance in parallel fashion so as to improve the efficiency and reduce the overall computation time. In the figure, Col1 to Col255 hold all the square of the horizontal distances from each received element to each pixel in the image. Firstly, the geometrical distances from the transmitter and the receiver to each pixel were calculated. The distance information in Col1 was combined with the corresponding y^2 data, followed by the square root operation; the distance from the pixel to the receiver can be calculated. Similarly, using the square of the horizontal distances, the y^2 data and the square root operation, the distances between the pixel and the transmitter can be evaluated as well. By superposing these two distances (transmitter to pixel and pixel to receiver), the total propagation distance could be evaluated. These propagation distances were then converted from IEEE754 data format to hexadecimal (hex) for easy addressing.

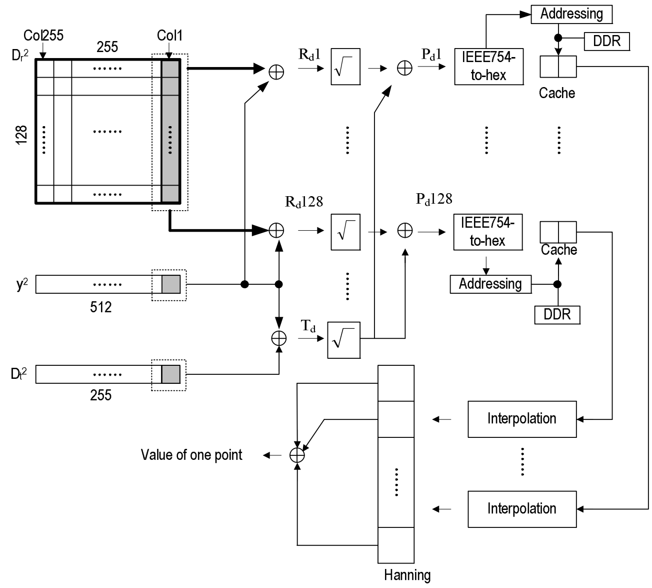

Fig. 3 Block diagram of parallel processing operation for each column

IV. SIMULATION RESULTS

To test the performance of the proposed system for the SA ultrasound imaging, a raw ultrasound echo signal was acquired. The parameters of the acquisition are listed in Table I. This ultrasound system consists of 97 transmitters, 128 receiving elements and the image size is 255×512. The proposed design for SA algorithm in FPGA was also implemented and simulated using MATLAB. By this method, it is convenient to verify the result with that of GPU. The results between MATLAB and GPU are matched well, and from Fig. 4, we can see that the image of the phantom with 3 holes is reconstructed correctly with only minor noise difference on the block.

Table I. Parameters for the imaging example

General imaging parameters	
Acoustic speed of tissue layer	1540*m/s*
Array size	128 elements
Array pitch	0.30*mm*
RF sampling frequency	40*MHz*
Imaging frequency	10*MHz*
Synthetic aperture imaging parameters	
Transmit aperture	97 channels
Receive aperture	128 channels
Transmit position	
-14.4*mm* ~ 14.4*mm* (spaced by 0.3*mm*)	
Beam-line imaging parameters	
Transmit aperture	97 channels
Receive aperture	128 channels
Number of beam-lines	127
Imaging depth	4*cm*

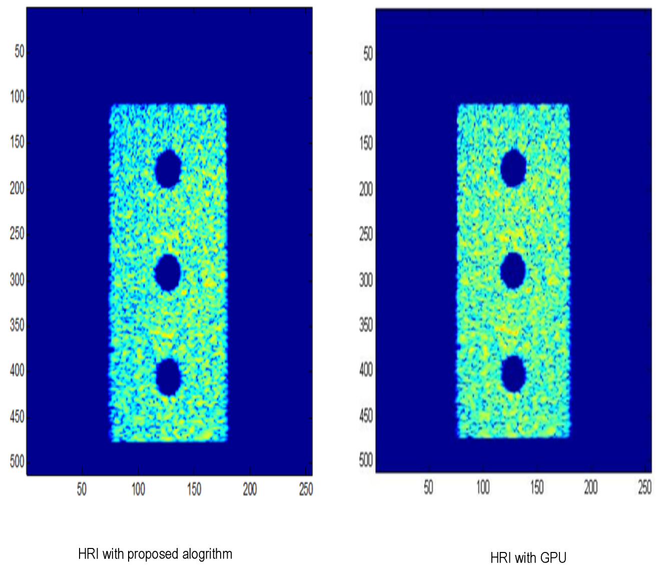

Fig. 4 Simulation results between MATLAB and GPU (255×512 pixels)

The algorithm was implemented with Verilog in FPGA and simulated in ModelSim. The process time for one frame is about 2.6*ms* based on Quartus II synthesis tool. From the synthesis of each module, it is found that the square root operation is the most time-consuming part. And this procedure can be optimized by replacing this built-in module with our later design, such as look-up table method.

V. Conclusion

The paper described a pipelined structure for the implementation of SA algorithm in ultrasound imaging. It can be found that current design can reduce the heavy computation time by using a parallel architecture. By the simulation and comparing with the result obtained by the GPU, the current design demonstrates the feasibility of implementing the real time high resolution ultrasound imaging. In the future, this algorithm will be further optimized and realized in custom design IC to minimize the power consumption and size. Then it will be applied for ultrafast ultrasound imaging systems.

Acknowledgment

This work was supported by The Science and Technology Development Fund of Macau (FDCT) under Grants 024/2009/A1, 087/2012/A3, and 047/2013/A2; The Research Committee of the University of Macau under Grants MYRG076(Y1-L2)-FST12-MPU, MYRG2014-00010-AMSV, MYRG079(Y1-L2)-FST12-VMI, MYRG103(Y1-L3)-FST13-VMI, and MRG014/MPU/2014/FST. The authors would like to thank Mr. Billy Y. S. Yiu and Dr. Alfred C. H. Yu of The University of Hong Kong for providing technical support and ultrasound data for the simulation.

References

[1] J. A. Jensen, "Medical ultrasound imaging," *Progress in Biophysics and Molecular Biology*, vol. 93, pp. 153-165, no.1, 2007.

[2] J. A. Jensen, S. I. Nikolov, K. L. Gammelmark, and M. H. Pedersen, "Synthetic aperture ultrasound imaging," *Ultrasonics*, vol. 44, pp. e5-e15, 2006.

[3] B. Y. S. Yiu, I. K. H. Tsang, and A. C. H. Yu, "GPU-based beamformer: Fast realization of plane wave compounding and synthetic aperture imaging," *IEEE Transactions on Ultrasonics*, *Ferroelectrics and Frequency Control*, vol. 58, pp. 1698-1705, 2011.

[4] M. Soumekh, "Synthetic aperture radar signal processing," *A Wiley-Interscience publication*, United States of America, 1999.

[5] J. J. Flaherty, "Synthetic aperture ultrasonic imaging systems," ed: Google Patents, 1970.

[6] P. Corl, G. Kino, C. DeSilets, and P. Grant, "A digital synthetic focus acoustic imaging system," in *Acoustical Imaging*, ed: Springer, 1980, pp. 39-53.

[7] M. O'Donnell and L. J. Thomas, "Efficient synthetic aperture imaging from a circular aperture with possible application to catheter-based imaging," *IEEE Transactions on Ultrasonics*, Ferroelectrics, and Frequency Control, vol. 39, pp. 366-380, 1992.

[8] S. I. Nikolov, K. L. Gammelmark, and J. A. Jensen, "Recursive ultrasound imaging," in *Proc. IEEE Ultrasonics Symp*., 1999, pp. 1621-1625.