VLSI Architecture for Ultrasound Array Signal Processor

Laseena C. A
Assistant Professor
Department of Electronics and Communication Engineering
Government College of Engineering Kannur
Kerala, India.

***Abstract*—A receive beamformer for high frequency linear ultrasound arrays has been implemented in FPGA. An efficient algorithm for Delay-and-Sum (DAS) receive beamformer is implemented. The system consists of 8 channels. The integer delays and fractional delay filter coefficients are calculated by MATLAB simulation. The sampling frequency is set as 100MHz. Radio frequency (RF) signals are digitized, delayed, and summed through a digital beamformer, which is implemented using a field programmable gate array (FPGA). The results showed that, for 367 echo samples the time required for beam-formation taking 50 scan lines is only 234.5µs.Hence this architecture can be used for real time B-mode imaging for Medical Ultrasound scanners.**

Keywords—Ultrasound, Beamforming, B-mode image, IP core, DAS.

I. Introduction

Array signal processing is to estimate the values of parameters by using available temporal and spatial information, collected through sampling a wave field with a set of antennas that have a precise geometry description. In medical ultrasound scanner, the ultrasound echoes received by the transducers determine the information about the properties of underlying tissues. The collected echoes are scaled and appropriately delayed to permit a coherent summation of the signals known as beamforming at the receiver. This new signal represents the beamformed signal for one or more focal points along a particular specific scan line. The beamformer operations are typically performed in application-specific integrated circuit (ASIC), field-programmable gate array (FPGA), DSP or a combination of these components.

This paper describes the VLSI architecture for a Delay-and Sum (DAS) receive beamformer for 8 array elements. The ultrasound echo signals are received from the Field II program procedures [1] and the final result was validated as B- mode image simulated in MATLAB.

A. Field II

Field II is a program for simulating ultrasound transducer fields and ultrasound imaging using linear acoustics. The programs use the Tupholme-Stepanishen method for calculating pulsed ultrasound fields. The Field program system uses the concept of spatial impulse responses. The acoustic transducer (as shown in Fig.1) on the left is mounted in an infinite, rigid baffle and its position is denoted by r_2. It radiates into a homogeneous medium with a constant speed of sound c and density r_0 throughout the medium. The point denoted by r_1 is where the acoustic pressure from the transducer is measured by a small point hydrophone. A voltage excitation of the transducer with a delta function will give rise to a pressure field, which is measured by the hydrophone. The measured response is the acoustic impulse response. Moving the transducer or the hydrophone to a new position will give a different response. Moving the hydrophone closer to the transducer surface will often increase the signal2, and moving it away from the center axis of the transducer will often diminish it. Thus, the impulse response depends on the relative position of both the transmitter and receiver (r_2–r_1) and hence it is called a spatial impulse response.

A perception of the sound field for a fixed time instance can be obtained by employing Huygens' Principle in which every point on the radiating surface is the origin of an outgoing spherical wave. The spatial impulse response is found by observing the pressure waves at a fixed position in space over time by having all the spherical waves pass the point of observation and summing them. The scattered field and received signal by the transducer is measured using the spatial impulse response, given by the equation (1), where δ(t) is the Dirac delta function.

$$h(\vec{r}_1,t) = \int_S \frac{\delta\left(t - \frac{|\vec{r}_1 - \vec{r}_2|}{c}\right)}{2\pi\,|\,\vec{r}_1 - \vec{r}_2\,|}\,dS \tag{1}$$

The received signal from the transducer is calculated as the spatial and temporal convolution of pulse-echo impulse, in-homogeneities in the tissue and pulse-echo spatial impulse response. Where, pulse- echo impulse is the convolution of transducer excitation, and the electro-mechanical impulse response during emission and reception of the pulse

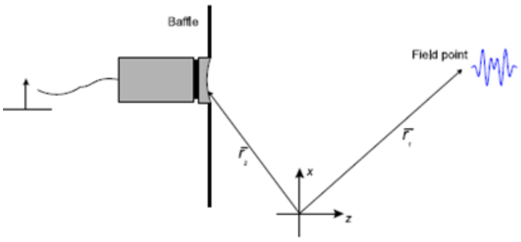


Fig. 1 A linear acoustic system

B. *Delay and Sum Beamformer*

Beamforming or spatial filtering is a signal processing technique used in sensor arrays for directional signal transmission or reception. In ultrasound scanner, beamforming is done both at transmitting and receiving ends. This is achieved by combining elements in a phased array in such a way that signals at particular angles experience constructive interference while others experience destructive interference. The main blocks of conventional Delay-Sum beamformer at the receiver end of the ultrasound system are shown in Fig. 2.

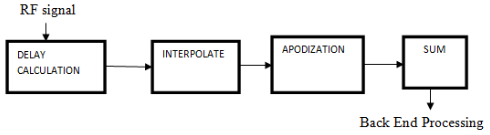


Fig. 2 Block diagram of Delay-and –Sum beamformer

1) Apodization

Apodization is done by applying weighing function to the aperture. The weights are independent of input data. This is done to reduce the side lodes and thereby increase the main lobe strength. The width of the window function is proportional to the depth to keep the width of the beam constant. After apodization, we get a narrow, directive beam pattern.

2) Delay calculation

An ultrasound linear phased array transducer contain over hundred transducer elements that may be multiplied and/or electronically steered and focused via phased array technique. Phase steering is accomplished by sequentially pulsing the array elements by calculation the inter – element delay using the equation (2),

$$\Delta\tau = \frac{d \sin \theta_s}{c} \qquad (2)$$

where $\Delta\tau$ is the time delay between adjacent element, d the distance between element, the θ_s required steering angle, and c is the wave speed(1540m/s) in the medium. The ultrasound linear phased array transducer model stated in this paper is able to simulate the pressure field for steering and focusing within the transition zone (near field), and beyond (far field). The focusing delay is calculated using the generalized focusing formula (3) for any number of elements N as stated in equation given below.

$$t_n = \frac{F}{c}\left\{\left[1+\left(\frac{\overline{N}d}{F}\right)^2+\frac{2\overline{N}d}{F}\sin\theta_s\right]^{1/2} - \left[1+\left(\frac{(n-\overline{N})d}{F}\right)^2-\frac{2(n-\overline{N})d}{F}\sin\theta_s\right]^{1/2}\right\} \qquad (3)$$

where t_n is the required time delay for element n=0,...N-1,$\overline{N}$ =(N-1)/2,d the center – to –center spacing between elements, F the focal length from the center of the array, θ_s the steering angle from center of array ,and c is the wave speed.

A fractional delay filter is used for the band-limited interpolation between samples. A discrete time signal x(n) is delayed to yield an output y(n) as y(n)=L{x(n)}=x(n-D),where L is a linear operator and D is a positive real number that can be split into the integer and fractional part as *D*= Int(*D*) +*d*. [3]In ideal case, when the desired delay D assumes an integer value, the impulse response of the signal reduces to a single impulse at n=D, but for non integer values of D the impulse response is an infinitely long, shifted and sampled version of the original signal. Thus an approximating (real-coefficient) filter must be implemented to solve this problem. There are several methods proposed in array signal processing for the implementation of fractional delay filter. Here, for the design of Rx beamformer, a Farrow structure fractional delay FIR filter with MMSE interpolator is used.

3) Interpolation

Fractional delay interpolation is used to generate fine delay. Wiener –Hopf Filter is used as the FIR filter for implementing fractional delay. [4] Wopt=Rx-1rxd.where Rx is the input autocorrelation matrix and rxd is the cross correlation vector. In this paper FIR filter with length 2 is considered. Thus

$$R_x = \begin{bmatrix} r(0) & r(1) \\ r(1) & r(0) \end{bmatrix}$$

where r(t) represents the autocorrelation function of X(n). If f_2 and f_1 represents upper and lower cut-off frequencies of the bandpass signal X(n), the R_x is calculated as:

$$R_x = \begin{bmatrix} 1 & \cos(w1)\sin(w)/w \\ \cos(w1)\sin(w)/w & 1 \end{bmatrix}$$

where w1= $\pi(f_1-f_2)/f_s$ and w= $\pi(f_2+f_1)/f_s$, with f_s as the sampling frequency. The cross correlation vector r_{xd} = $[p(0)\ p(-1)]^T$, where p(0) and p(-1) represents the cross correlation function between d(n) and X(n) for lags of 0 and -1 respectively; and calculated as
p(0)= cos(w4).*sin(w3+eps)/(w3+eps)
p(-1)=cos(w1-w4).*sin(w-w3+eps)/(w-w3+eps)
where w3= w x *del* ,w4=w1 x *del* ,eps is the floating point accuracy and *del* is the fractional part of the delay calculated by the equation (3).

4) Sum

Finally the delayed signals are summed and beamformed output is obtained.

II. MATLAB MODELING

The VLSI architecture design implemented in this paper is based on the algorithm developed and verified by MATLAB model simulation. Field II procedures are used to design the transducer array and generate pulse-echo signals received by individual transducer elements.

C. *Transducer array design*

An ultrasound array consists of number of transducer elements arranged in different methods to form linear, convex, annular, and phased arrays. Here a linear phased array transducer is designed with number of elements as 64. The element width, kerf, height, focus and element subdivisions are designed as per the procedures of Field II and as shown in TABLE 1. Speed of sound in tissue is taken as 1540m/s and sampling frequency as 100MHz.

D. Delay and Sum Beamformer design

Using Field II and MATLAB simulations , an algorithm for Delay-Sum beamformer is designed. Transducer array designed using Field II is used to generate pulse echo signals at focus and a beamformed signal is formed. The RF data obtained is undergone envelope detection, log compression and scan mode conversion to view the data as a B-mode image. In order to implement the delay sum beamformer algorithm, eight echo signals are generated using FIELD II The delay calculation for linear phased array is carried out using the equation (2) and a combination of coarse delay and fin delay is obtained. After performing apodization, integer delay is given to individual signals. Then the signals are undergone fin delay filtering by the pre calculated filter coefficients. Finally obtained delayed signals are summed. The RF data obtained is viewed as the B-mode image and compared with the previous image formed using Field II beamformer output. As the two images were identical, the calculated integer delays and fractional filter coefficients are stores as text files for the further processing for the implementation of VLSI architecture.

Table 1 Array Specifications

Array type	Linear focused array
Number of physical elements	8
Element height	13mm
Element width	0.265 mm
Element kerf	0.025 mm
Element pitch	0.290 mm
Elevation lens focus	60 mm
Emitter focus	[20 0 20] mm
Element sub division in x-direction	1
Element subdivision in y-direction	15
Transducer center frequency	3.5 MHz

III. ARCHITECTURE DESIGN

In VLSI architecture design of array signal processor, bottom-up approach is adopted. Initially sub architectures are designed and finally all of them are integrated to give adaptive beamformer output. FPGA implementation is done with reference to the algorithm that have been developed and verified in the simulation model of MATLAB. VLSI architecture consists of delay generation blocks, memory blocks and controller. Controller generates control signals for the synchronization of all these blocks.

A. High level Architecture

The high level architecture as shown in Fig.3 consists of the controller and different modules of Delay-and-Sum beamformer. The controller controls the whole processing and is activated by main clock, reset and START_READ signals. Here 367 samples of individual echo signals are generated using FIELD II and are stored in 1.15 formats. These signals are loaded in eight separate DPRAM (dual port RAM) as coe files. The coarse delay and fine delay filter coefficients, calculated in MATLAB modeling are also loaded in DPRAM. The controller is designed as FSM with states as shown in the data flow diagram. As the START_READ signal is given, the individual samples from each DPRAM are passed through the apodization module. Apodization is done using hanning window function. Here the hanning window function coefficients are loaded as coe file and in multiplied with the individual samples coming from DPRAM. The result is the moved to coarse delay module and fine delay module. Finally obtained delayed signals are summed and stored in another DPRAM. The implementation of each block in this architecture is stated in detail.

B. Low level Architecture

Low level architecture consists of five sub architectures. They are controller, Apodization module, coarse delay module, fine delay module and sum module. DPRAM modules are used to read and write the echo signals, delay, filter coefficients and Beamformed output.

1) Apodization module

Here, the individual samples from DPRAM are multiplied with the hanning window coefficients. There are eight apodization modules instantiated for the design. The IP core multiplier is implemented as apodization module as shown in Fig. 4. The output is valid and moved to next block only if APO_OUT_VALID bit become high

2) Course delay module

Coarse delay or integer delay is calculated in Matlab determines, the new address location of the incoming samples. Thus the integer delay is added with the corresponding address of DPRAM . There are eight coarse delay modules. IP core adder is used to sum the two values and the result is valid only when COARSE_OUT_VALID is high.

3) Fin delay module

Fin delay or fractional delay is implemented using two tapped farrow structure .The filter coefficients are calculated using MMSE method. All the calculations done in Matlab are stored and loaded into this module. The delayed signals are to be stored sample by sample into the corresponding address location specified by the OUT_ADDR of coarse delay module. Here the output pin FIN_OUT_VALID becomes high if each sample gets delayed. There are eight fin delay modules in the design. Each one consists of DFF_DELAY module, two multiplier IP core and one ADDER IP core.

4) Write delay signal module

This module consists of an array of 10,000 data size. When the FIN_OUT_VALID is high, the write-delay module is enabled and the DATA_OUT values of fin module is stored in to the memory location specified by OUT_ADDR pin of coarse delay module. When all the 367 delayed samples are written in this array, output pin WRITE_COMP becomes high, and the data is made available for further processing. To write the delayed eight signals, we used eight write-delay-modules in this design.

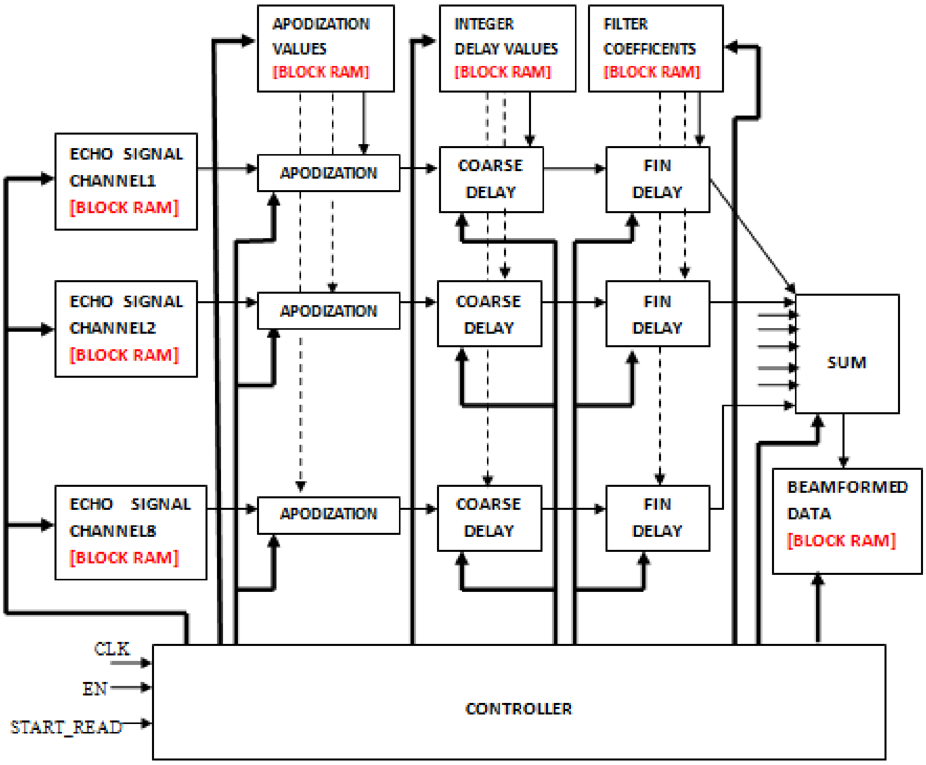

APODIZATION VALUES [BLOCK RAM]
INTEGER DELAY VALUES [BLOCK RAM]
FILTER COEFFICENTS [BLOCK RAM]
ECHO SIGNAL CHANNEL1 [BLOCK RAM]
APODIZATION
COARSE DELAY
FIN DELAY
ECHO SIGNAL CHANNEL2 [BLOCK RAM]
APODIZATION
COARSE DELAY
FIN DELAY
SUM
ECHO SIGNAL CHANNEL8 [BLOCK RAM]
APODIZATION
COARSE DELAY
FIN DELAY
BEAMFORMED DATA [BLOCK RAM]
CLK
EN
START_READ
CONTROLLER

Fig. 3.High Level Architecture

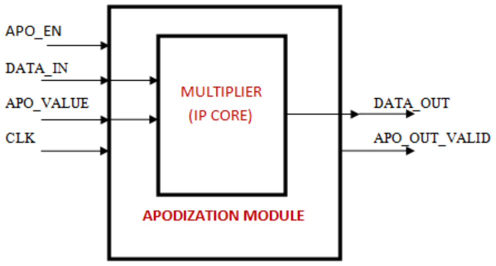


Fig.4 Apodization module

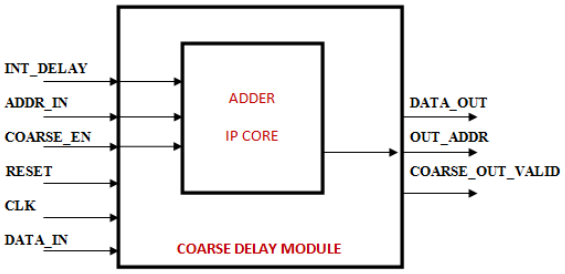


Fig.5 Coarse delay module

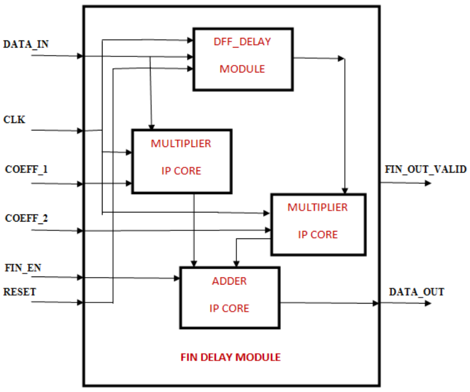


Fig. 6 Fin delay module

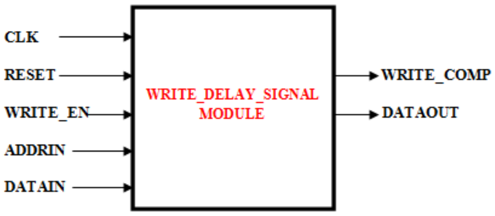


Fig. 7 Write delay signal module

5) Sum module

The eight delayed signals are summed here. Sum module is enabled only when the WRITE_COMP pin of all the write-delay-module become high. When all the signals are summed, SUM_OUT_VALID becomes high and the SUM_OUT will give the beamformed data. In the design, the final data is stored in another DPRAM. As the entire data is written, WRITE_BF_COMP becomes high. After that the whole processes repeated by moving the controlling action to initial state.

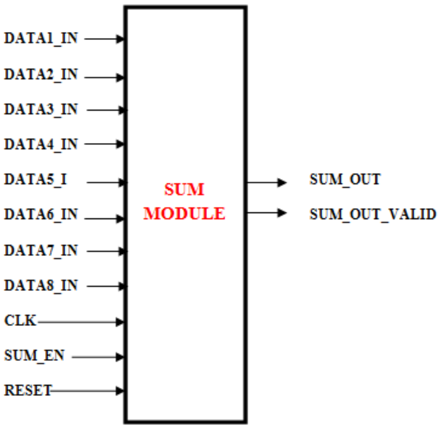


Fig 8. Sum module

6) DPRAM

There are twelve Dual-port RAM modules used in this VLSI architecture design. It is implemented from logic IP Block memory generator. Dual Port RAM is a memory which has dual ports, one port used to write the data into memory and other port used to read the data from memory as shown in Fig 9.Timing diagram for READ and WRITE process using this module is shown in Fig. 16.

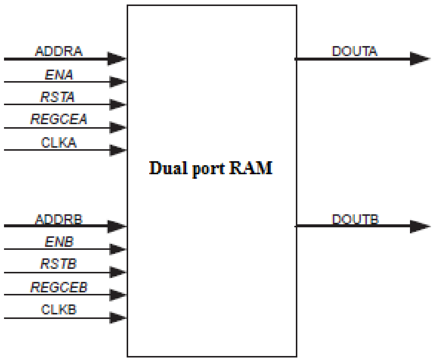


Fig.9 DPRAM

7) Controller

As mentioned earlier, controller is designed as a FSM, which is activated by system clock signal and reset signal. The entire program starts when START_READ is high. All the echo sampled and stored echo signals are simultaneously read from the lock memory. Based on the output valid signals of above discussed modules, all the controls signals are generated by the controller. The control actions performed cause the flow of data from one module to another. Finally, after writing the beamformed data into block memory, WRITE_BF_COMP signal is generated. Then the controller action is moved to initial state and the entire read operation is repeated for next RF data.

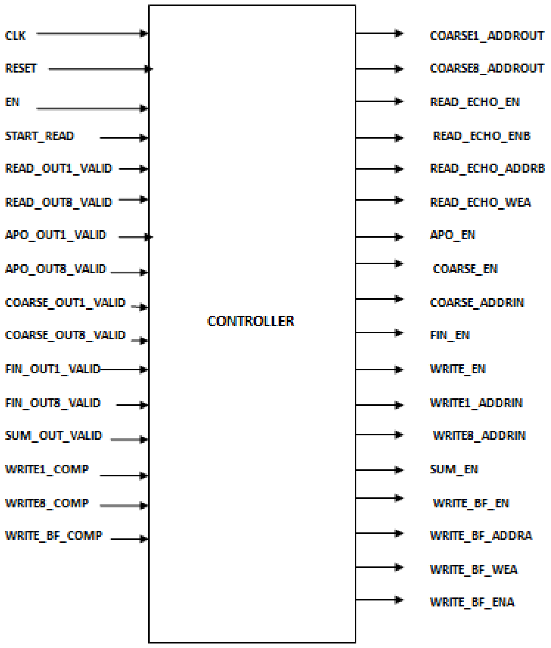


Fig. 10 Controller

IV. IMPEMENTATION

The architecture design is based on the simulation results of Matlab modeling. The implementation is done as per the flowchart shown in Fig.11. Initially, linear phased array is designed and implemented using Field II. The pressure field calculations and its variations on changing focus are obtained. To design the conventional Delay-and-Sum beamforming, at first pulse echo signals received by individual array elements are collected. Using array parameters and generalised focusing equation, time delay for individual elements are calculated. Again, the delay is divided as integer delay and fractional delay. Interpolation filter coefficients are calculated in Matlab. After windowing individual echo signals with hanning function, integer delay is applied. The resultant signals are interpolated using fractional delay filtering. After all these processes, the signals are summed to obtain the beamformed data. The RF data is envelope detected, log compressed and scan converted to obtain a gray scale B-mode image. Obtained image is compared with the B-mode image formed using Field II. Virtex6 FPGA was selected as the platform for implementation.ML605 evaluation board, was used as the hardware as XC6VLX240T-1FFG1156 FPGA is the target device. This device was selected because of its high speed performance required for real time implementation.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 5 Issue 01, January-2016

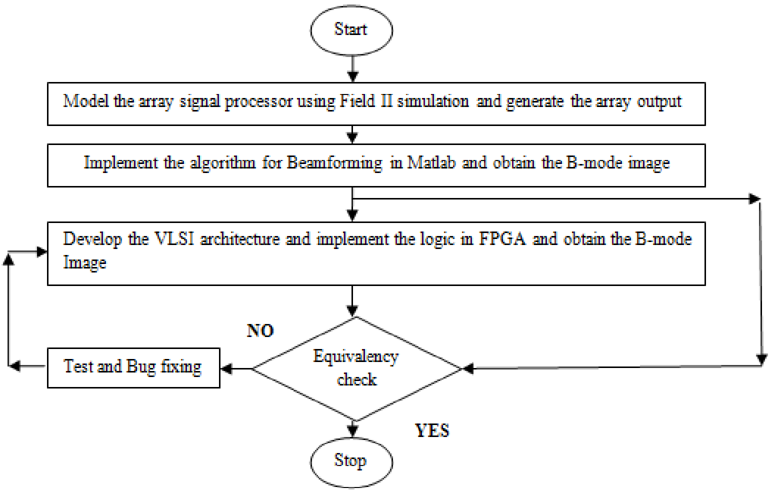
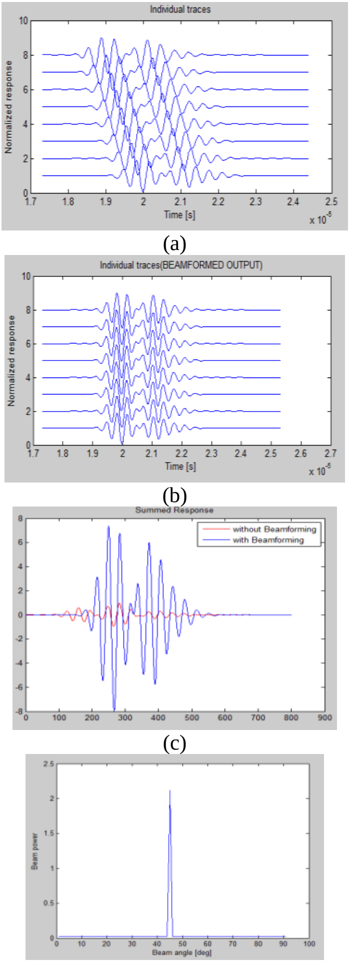

Start
Model the array signal processor using Field II simulation and generate the array output
Implement the algorithm for Beamforming in Matlab and obtain the B-mode image
Develop the VLSI architecture and implement the logic in FPGA and obtain the B-mode Image
NO
Equivalency check
Test and Bug fixing
YES
Stop

Fig. 11 Implementation program flow

V. RESULTS AND DISCUSSION

In Matlab model implementation, the individual echo signals are obtained using Field II procedure. The results obtained in designing Delay-and –Sum beamformer in Matlab are shown in Fig. 12 and 13.



(a)

(b)

(c)

(d)

Fig.12: (a) Signals of individual elements.(b) Delayed signals of individual elements, (c)Summed response of individual signals with and without beamforming, (d) beam power plot showing maximum power only at 45°.

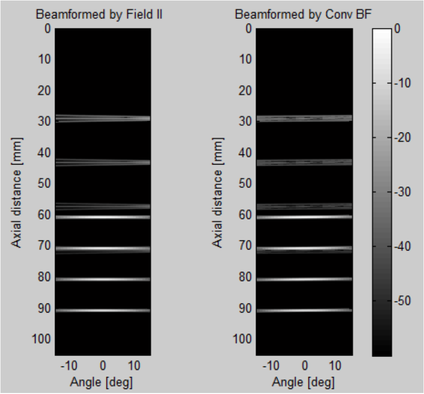


Fig.13:Comparison of Beamformer data obtained by Field II and Delay-Sum-Beamformer

Results of ISE simulation of fin delay module is compares with that of MATLAB code as shown in Fig 14.The delays and filter coefficients are loaded in DPRAM and the read-write timing diagram is obtained as shown in Fig.15. Controller is implemented as FSM with a lot of control signals as shown in Fig.17. The final bit file generated was loaded in ML605 evaluation board. By using UCF, the device was activated. The power analysis report obtained, the total power requirement for Porting architecture is 0.4793 W and for Clocks it is 0.02210 W.The timing analysis obtained is shown in Table 3.

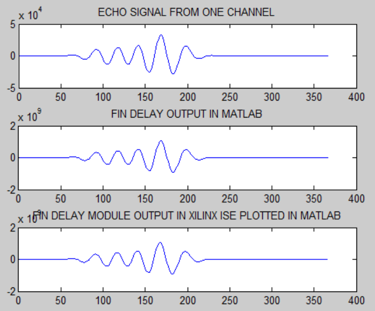


Fig. 14 Comparison of Fin delay module output

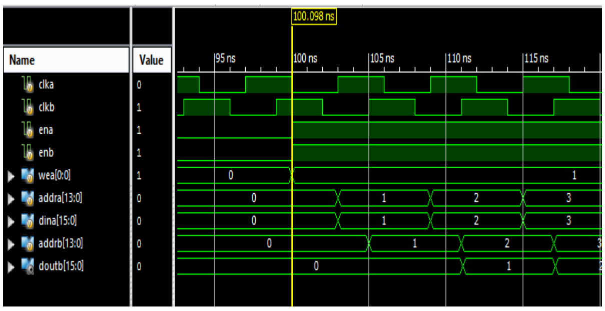


Fig.16 Timing diagram of WRITE – READ operation performed in block memory

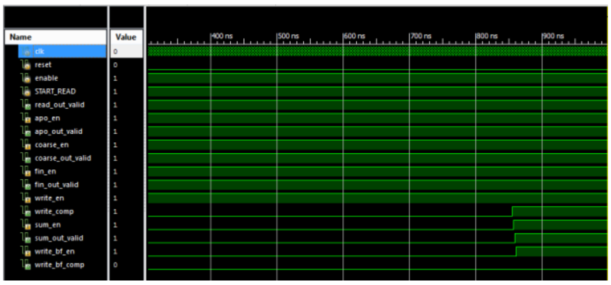


Fig. 17 Timing diagram of control signals

Table 2
Device Utilization Summary

Slice Logic Utilization	Used	Available	Utilization
Number of Slice Registers	839	301,440	1%
Number of Slice LUTs	492	150,720	1%
Number of fully used LUT-FF pairs	417	890	46%
Number of bonded IOBs	37	600	6%
Number of RAM/FIFO	8	416	1%
Number of BUFG/BUFGCTRLs	3	32	9%
Number of DSP48E1s	16	768	2%

VI CONCLUTIONS

Initially beamformer algorithm is applied in MATLAB. It include mainly designing appropriate apodization, integer delay, fin delay and finally summing up of all signals. Thus the FPGA implementation also required the study various low level architectural designs. All the process is divided in to different states and thud a controller is designed for overall synchronization. The entire architecture is implemented as an efficient design in vertex6 platform with minimum number of devices. This design is carried out for eight channels. The power consumption observed is very low and the total time required for beamforming is only 234.5µs. Future work includes, upgrading the number of elements from 8 to 64, and improving the design for real time imaging.

Table 3 Timing Analysis

Minimum period	6.291ns
Maximum Frequency	158.945MHz
Maximum output required time after clock	0.37ns
Maximum combinational path delay	No path found
Number of clock signals required to make WRITE_BF_COMP high	746
Time take for beamforming along one angle	4.69µs
Time take to complete 50 scan lines	234.5 µs

REFERENCES

[1] Jorgen Arendt Jensen “Ultrasound imaging and its modelling” Chapter of the book *“Imaging of Complex Media with Acoustic and Seismic Waves”,* published by Springer Verlag, 2000.

[2] L. Azar, Y. Shi and S. C. Wooh;“Beam focusing behavior of linear phased arrays”, NDT&E International, vol.33.page 189–198, July 2000.

[3] Timo I. Laakso,Vesa Valimaki,Matti Karjlainen and Unto K. Laine, “ Splitting the unit delay: tools for fractional delay filter design” IEEE Signal Processing Magazine page 30 – 60 January1996.

[4] S. Sami Deeb and Robert A. LaTourette, “Derivation of Beam Interpolation Coefficients with Application to the *K-ω* Beamformer” *NUWC-NPT Technical Report 11*,287 15 June 2001.

[5] Toby Haynes, “A Primer on Digital Beamforming”,*Spectrum Signal Processing*, March-26-1998.
http://www.spectrumsignal.com.