Ultrasound B-Mode Back End Signal Processor on FPGA

Jayaraj U K
Scientist / Engineer 'C'
NIELIT, Calicut

S. Gayathri
Associate professor
Dept.of Electronics and Communication Engineering
Sri Jayachamarajendra College Of Engineering
Mysoru

Shamya A
Dept. of Electronics and Communication Engineering
Sri Jayachamarajendra College Of Engineering
Mysoru

Abstract— **Ultrasound imaging is a non invasive medical diagnosis technique .The back end system is the core of the imaging system. The performance of back end system mainly depends on the signal processor. This paper presents a back end signal processor for high frequency high frame rate ultrasound B mode imaging. Processor execute the RF data out from the digital beam former.**

It receives the useful information from the acquired echo signal and displaying that information on a monitor.

Signal reconstruction is done by Hilbert Transform Technique that is implemented using finite impulse response filters (FIR) applied on each scan line returning from the beam former. Image reconstruction is done through scan conversion and interpolation modules.

The signal processor is implemented on single FPGA. The processor architecture consists of RAM ,FIR filter, adder CORDIC block. The processor output is compared and validated with the model using MATLAB$^{@TM}$. The design is applied on Xilinx Vertex 6 FPGA.

Keywords— ***MATLAB,CORDIC,FIR filter, RAM, I-in phase, Q-quadrature , FPGA***

I.INTRODUCTION

High frequency ultrasound transducers have made it possible to open up new applications including ophthalmic, dermatologic, intravascular, and small animal imaging because of their fine spatial resolution on the order of several tens of micrometers. In addition to the fine spatial resolution, a fine temporal resolution (more than 200 images per second) is applicable to cardiac imaging of the mouse

where the heart rate is 5-10 beats per second. For these applications, two key elements are required: high-speed acquisition of echo signals (front-end system) and high-speed signal processing functions to extract clinically useful information from the acquired echo signals and display the information on a monitor in real time (backend system). It is a challenge to develop high-speed signal processing functions used in the backend system to support high frame rate imaging because these functions must be capable of providing very fast computational speed and wide bandwidth of data transfer between each functional block.

In the back end signal processing system contains the logical blocks such as dc canceller ,digital time gain compensator, a high speed envelope detector,
a high speed digital scan convertor and display module[1].

In the developed processor architecture includes the envelope detector ,dynamic range compression and display module logic blocks.

II.THEORY

A block diagram illustrating the data flow in our proposed processor architecture is shown in Fig.1 .

RF signal after beam forming is taken as a signal input to the back end system. Signal reconstruction is done by Hilbert Transform Technique that is implemented using finite impulse response filters (FIR) applied on each scan line returning from the beam former [2]. Dynamic range of the RF data reduced logarithmically. Image reconstruction is done through scan conversion and interpolation
modules [2].Block schematic shown below in figure.1

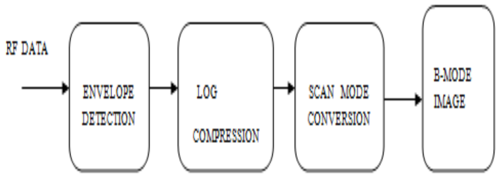


Figure.1 Block schematic of functional units of the Back end signal processor

III ARCHITECTURE

High level architecture for processor is developed and its block schematic shown in Figure.2

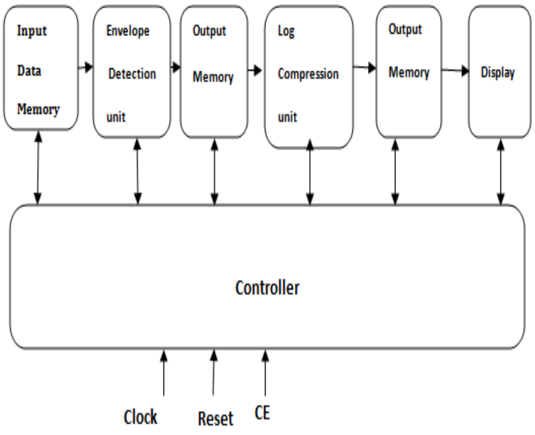


Figure.2 Schematic of Back end signal processor architecture.

Input data for the processor is RF data, generated using field-II. Field II[9] 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 program is capable of calculating the emitted and pulse-echo fields for both the pulsed and continuous wave case for a large number of different transducers. Also any kind of linear imaging can be simulated as well as realistic images of human tissue. The program is running under MATLAB$^{@TM}$ on a number of different operating system (Windows, Linux, Mac OS X).

The data is stored in the memory. The output from memory acts as input to envelope detection block. The envelope detected data is saved in the envelope data out memory. The output from the memory is connected to log compression unit. The input memory, envelope detection and log compression units are controlled by a controller. The controller and above three blocks forms the back end signal processor functional units.

III.1 DETAILS OF BACK END SIGNAL PROCESSOR FUNCTIONAL UNITS

III.1.1 Envelope detection:

The analytic envelope of the signal is calculated as the square root of the sum of the squares of the real(In phase –I) and quadrature components(Q) [2]. There are certain techniques to get I and Q components of the RF signal. The commonly used methods are quadrature demodulation and Hilbert transform technique. Quadrature modulation approach requires extra circuitry[4].But Hilbert transform technique requires only memory and it is inexpensive. Therefore it is used in medical applications. The most accurate way of obtaining the quadrature components was to pass the echo signal through a hilbert transform, because it provides 90-degree phase shift at all frequencies.

Digital Hilbert transformers are a special class of digital filter whose characteristic is to introduce a $\pi/2$ radians phase shift of the input signal. In the ideal Hilbert transformer all the positive frequency components are shifted by $-\pi/2$ radians and all the negative frequency components are shifted by $\pi/2$ radians.

Hilbert transformers can be designed either as Finite Impulse Response (FIR) or as Infinite Impulse Response (IIR) digital filters [1], [2], and they are used in a wide number of Digital Signal Processing (DSP) applications, such as digital communication systems, radar systems, medical imaging and mechanical vibration analysis, among others [3].

IIR and FIR filters performs phase and magnitude approximation respectively.FIR filter is most preferred since its stability and insensitivity to coefficient rounding.

The linear time invariant (LTI) a FIR filter can be defined as

$Y(n)=b_0x[n]+b_1x[n-1]+b_2x[n-2]\ldots\ldots.b_m\, x[n-M]$---(1)

$$Y(n)=\sum_{n=0}^{M} bi.x[n-i].\qquad\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(2)$$

Where x is input signal
Y is output signal
b is filter coefficients

The designed FIR Hilbert filter can be used to generate the Hilbert transformed data of the received echo signal. The impulse response of the Hilbert filter with length N (odd number) is defined as

$$h[n]=\begin{cases}\frac{2}{\pi}\,\frac{\sin^2(\pi(n-}{n-\alpha} \\ 0\end{cases}\ldots\ldots\ldots\ldots\ldots\ldots..(3)$$

where α =

Selected filter order for the design is N=67 and hamming window is used to reduce the side lobe effects [2]

The low level Field programmable gate array (FPGA) architecture for envelope detection is shown in Figure 3.

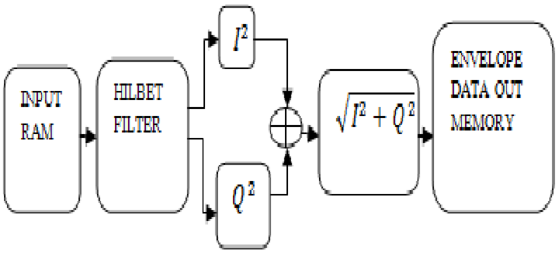


Figure.3: Low level architecture of envelope detection

The input data is stored in the input RAM. Data is represented in 1.15 fixed point binary number format. The data out from memory is given to FIR filter.FIR filter IP core is generated with Hilbert filter coefficients.

Filter frequency response shown in Figure.4 and characteristics is like band pass filter. Filter has I and Q output.

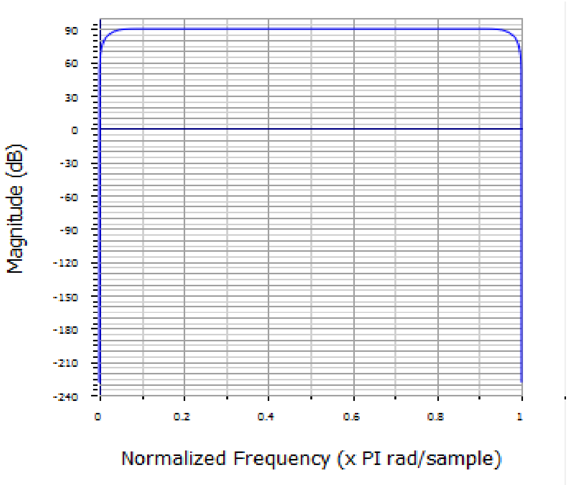


Figure.4: Frequency response of FIR filter

The input data and filter coefficients are represented in 1.15 fixed point number format. The output of FIR filter IP is in 2.30 fixed point number format. Q output data is converted to 1.15 fixed point formats by arithmetic right shift for equalizing the data format for square and sum logic. Sum of squares of I and Q computed using multiplier and adder unit. Square root of adder data output is computed using Xilinx FPGA's coordinating rotation digital computer (CORDIC) IP core. The output of CORDIC IP is stored in envelope data out memory.

III.1.2 Log compression

After the envelope detection the signal values can be rescaled to fit the grayscale color map in order to display the signals as a B-Mode image. But usually the image will be very dark and structures will be hard to see.

This is due to the fact that there are usually some relatively high peaks in the envelope signal corresponding to materials with a very high reflectivity.

When rescaling the whole envelope this peaks cause a down-shift of the other values. As a result the image looks dark. To compensate for such peaks the signal values are not rescaled linearly. There are various possibilities to rescale the image non-linearly. All of them have in common that higher values are more attenuated than lower ones in order to increase the overall brightness while preserving the order of the signal values.

Two popular methods are calculating the square root or the logarithm of the envelope signal before rescaling the values. The derivatives of both functions decrease with high values and therefore high values are more attenuated. Usually the value range is compressed to 8 bits since the ability of the human eye to distinguish grayscale colors is limited anyway Low level FPGA architecture for Log compression is shown in figure 5.

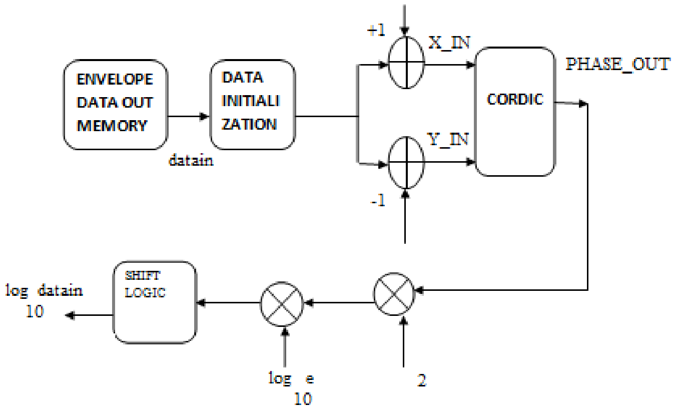


Figure.5: Low level architecture of log compression

The methods to compute the logarithmic values of data using digital circuits are look up table based algorithms and iterative method.[6].Look up table method is faster ,and simple. but it is useful for small size data. Iterative method doesn't require large memory and has high precision. [6]Taylor's series expansion is most post popular method to manually compute logarithms .It requires many multipliers and adders. Also it has slow convergence.[5] [6]

CORDIC algorithm is more suitable for hardware implementation since it requires adders and multipliers.[5][6][7]

Calculation of $\log_{10}$ using the CORDIC algorithm depends on inverse hyperbolic tangent modes of CORDIC and the use of formulae [8] for conversion between logarithm base-10and natural logarithm

$$l_n(m)= 2*\tanh^{-1}(m-1/m+1)\ldots\ldots\ldots\ldots\ldots\ldots\ldots(4)$$

$$\text{Log}_{10}(m)=l_n(m)*\log_{10}(e)\ldots\ldots\ldots\ldots\ldots\ldots\ldots(5)$$

The inverse hyperbolic tangent is computed by calling the CORDIC IP core of Xilinx ISE software. The data input for IP core is in signed fraction format and it is expressed in fixed point 2's complement number with an integer width of 2 bit. PHASE_OUT is represented in fixed point 2's complement number with an integer width of 3bit .Multiplication PHASE_OUT is carried out by arithmetic left shift by 1bit.The shifted data is multiplied with log10(e).

III.1 3.Controller

Controller will control the functional units in the processor which is designed using finite state machine model. Controller FSM is shown below

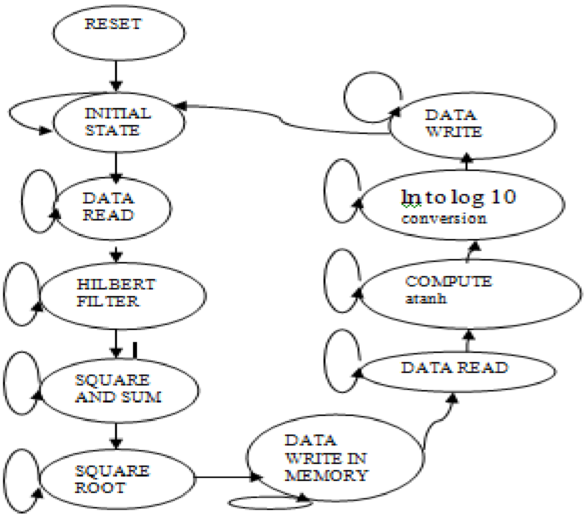


Figure.6 Controller FSM

The image data of cyst phantom generated using Field II is loaded into the input memory as . coe file. Image data consists of 565 samples per each scan line .Total number of scan lines for -35 ° to 35° viewing angle is 104 .Total number of samples in the image data is 58760.Data write width of input in memory is 16bit and data write depth is 58760.

Each state has a data valid signal. If the valid signal is high controller switches states from current state to next state. Otherwise the current state will wait till valid signal high.

If the data write width of output memory reaches last count value , the whole compressed data is saved in the memory. The output memory is connected to display unit to monitor the image.

IV. SIMULATION

The phantom data that was used to generate results is cyst phantom. The parameter set for the probe to acquire RF data is shown in table 1

Parameter	Type/value
Sampling frequency	40MHz
Transducer center frequency	2.5MHz
Speed of sound	1540
Element height	13mm
pitch	0.29mm
Kerf	0.025mm
Transmit and receive apodization	Hanning window
Oversampling ratio	4
Number of elements	64
Transducer type	Phased array

Table.1 Probe specifications

Using field II, with the computer phantom the beam formed data is generated. This data is used for modeling the back end processor .Design of low level architectures of envelope detection and dynamic range compression block is modeled in MATLAB$^{@TM}$ and validated the results from FPGA

The FPGA output is compared with MATLAB$^{@TM}$ output. Simulation results of envelope detection in MATLAB$^{@TM}$ and FPGA is shown in figure. 7 and figure 8.Log compression results are shown in figure 9 and figure 10.

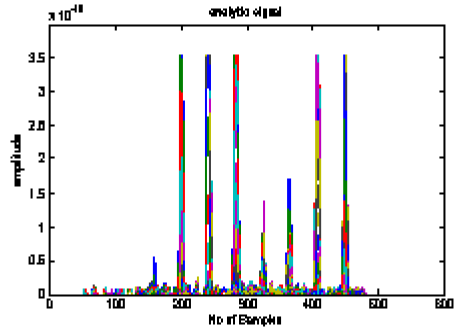


Figure.7 FPGA out

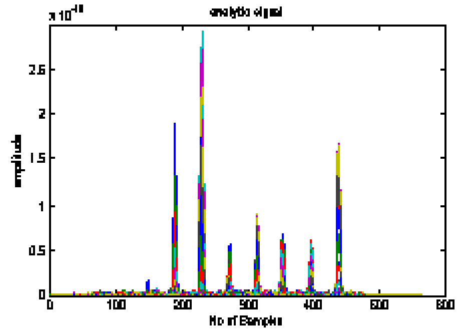


Figure.8 MATLAB$^{@TM}$ output

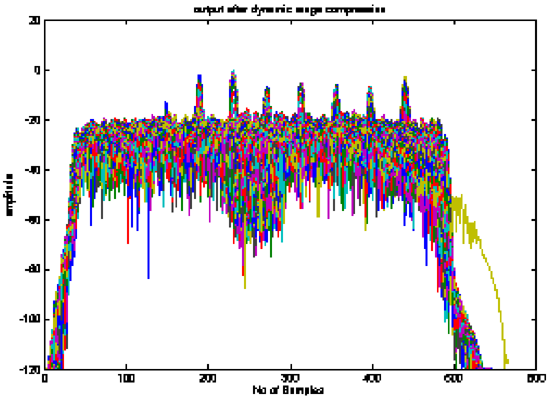


Figure.9.Log compression MATLAB$^{@TM}$ output

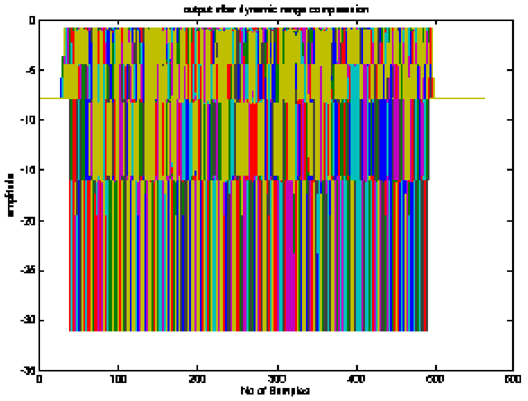


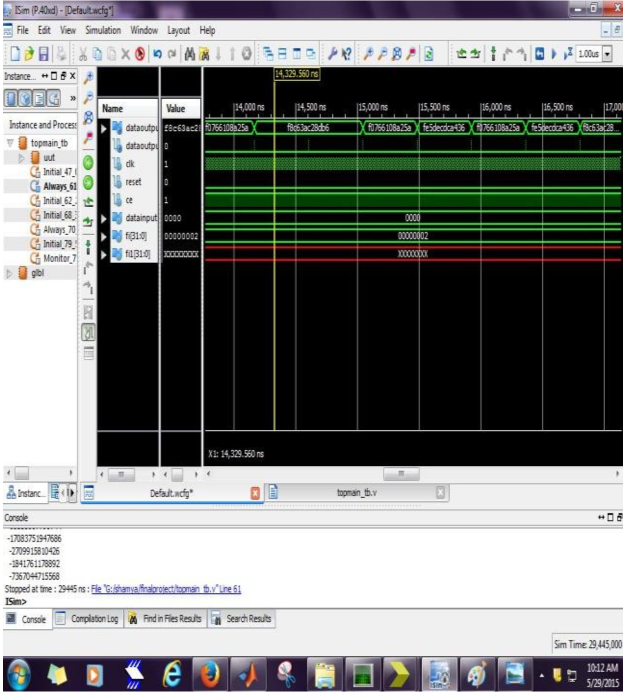
Figure.10.Log compression FPGA output

IMPLEMENTATION

The back end signal processor is implemented on Virtex-6 XC6VLX240T-1FFG1156 FPGA. The verilog code is synthesized using Xilinx ISE 14.3 simulator. The RTL schematic obtained after synthesis is shown in Figure 11.Device utilization summary is shown in Table2.After synthesis run the design implementation. Implementation steps consists of translate, Map, Place and route and programming file generation

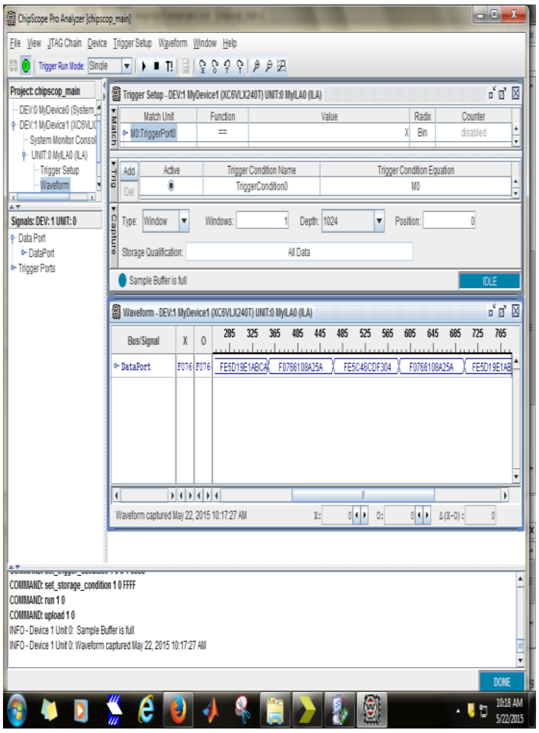
- Translate, which merges the incoming netlists and constraints into a Xilinx® design file
- Map, which fits the design into the available resources on the target device
- Place and Route, which places and routes the design to the timing constraints
- Programming file generation, which creates a bitstream file that can be downloaded to the device

Chip Scope Pro analyzer is used to test our design in hardware .Chip scope Pro Analyzer is a GUI which allows the user to configure device ,choose triggers, setup the console and view results of the capture on the fly. In order to use the chip scope Pro add Chip scope Pro core into the design using core generator. Simulation and Chip scope Pro results are shown in Figure 11 and Figure12.



```
-17083751947686
-2709915810426
-1841761178892
-7367044715568
Stopped at time : 29445 ns : File "G:/shamya/finalproject/topmain_tb.v" Line 61
ISim>
```

Figure 11.ISIM sceen shot


ChipScope Pro Analyzer [chipscop_main]
File View JTAG Chain Device Trigger Setup Waveform Window Help
Trigger Run Mode: Single
Project: chipscop_main
Trigger Setup - DEV:1 MyDevice1 (XC6VLX240T) UNIT:0 MyILA0 (ILA)
Match Unit
Function
Value
Radix
Counter
M0:TriggerPort0
Bin
disabled
Trigger Condition Name
Trigger Condition Equation
TriggerCondition0
M0
Type: Window
Depth: 1024
Storage Qualification: All Data
Sample Buffer is full
IDLE
Waveform - DEV:1 MyDevice1 (XC6VLX240T) UNIT:0 MyILA0 (ILA)
Bus/Signal
DataPort
Waveform captured May 22, 2015 10:17:27 AM
DONE

```
COMMAND: set_storage_condition 1 0 FFFF
COMMAND: run 1 0
COMMAND: upload 1 0
INFO - Device 1 Unit 0:  Sample Buffer is full
INFO - Device 1 Unit 0: Waveform captured May 22, 2015 10:17:27 AM
```

Figure 12.Chipscope screen shot

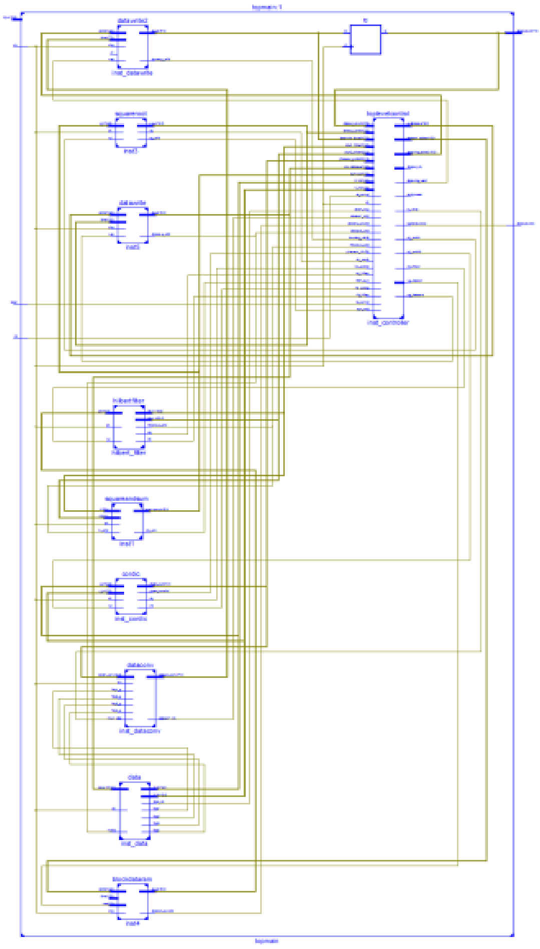


Figure.11.Internal RTL schematic of Proposed ultrasound B-mode back end signal processor

V. RESULTS

The individual module output from MATLAB$^{@TM}$ and FPGA is compared and validated the results. Cyst phantom output image using MATLAB$^{@TM}$ and FPGA.

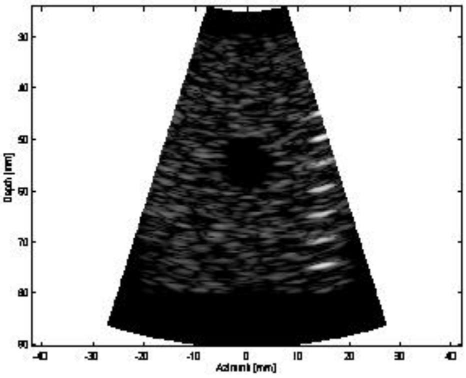


Figure 13.Output image from Cyst phantom obtained in MATLAB$^{@TM}$

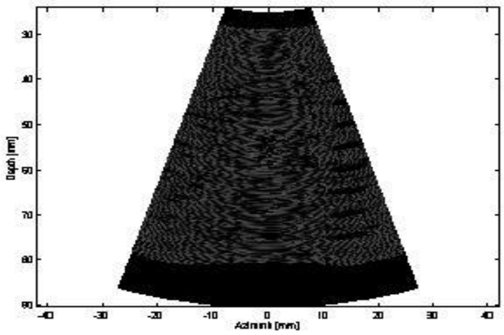


Figure 14.Output image from Cyst phantom obtained in FPGA

The hardware architecture of ultrasound B-mode back end signal processor implemented on Vertex6 FPGA evaluation board . Device utilization summary is shown in the table 2.

DEVICE UTILIZATION SUMMARY

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice Registers	1267	301440	0%
Number of Slice LUTs	2190	150720	1%
Number of fully used LUT-FF pairs	883	2574	34%
Number of bonded IOBs	52	600	8%
Number of Block RAM/FIFO	134	416	32%
Number of BUFG/BUFGCTRLs	1	32	3%
Number of DSP48E1s	21	768	2%

Table 2.Device utilization summary

CONCLUSION

This paper demonstrates the architecture of back end signal processor and its functional units. The verilog code for the processor is simulated using Xilinx ISE$^{@TM}$ .The code is synthesized and ported the RTL implementation in Xilinx Vertex6 ML605 evaluation platform. The processor hardware output is compared with the MATLAB$^{@TM}$ model. The MSE error between the two models is 0.2%.Maximum clock frequency is 138.485MHz.

Future works related to this is evaluating the performance of processor by adding pipelining in each stage in processor units.

REFERENCES

[1] Jin Ho Chang, Jesse T. Yen, Lei Sun, and K. Kirk Shung , “Backend Processing System for High-Frequency,High-Frame Rate Ultrasound B-mode Imaging”, NIH Resource Center for Medical Ultrasonic Transducer Technology, Department of Biomedical Engineering

[2] Mawia Ahmed Hassan,Yasser Mostafa Kadah ,”Digital signal processing methodologies for conventional digital medical ultrasound imaging system”, *American Journal of Biomedical Engineering* 2013, 3(1): 14-30 DOI: 10.5923/j.ajbe.20130301.03

[3] David Ernesto Troncoso mero and Gordana Jovanovic olecek ,”Digital FIR Hilbert Transformers :Fundamentals and Efficient Design Methods”

[4] Engy R.EL Mahdy, Rania M. Khalil, Shaimaa Sherif, and Dr.Yasser M. Kadah,” FPGA Based Backend System for Medical B-Mode Ultrasound Imaging**”,** *proc. Cairo international biomedical engineering conference* 2006

[5] Liu Bangqiang,He Ling,Yan Xiao,”Base –N logarithm Implementation n FPGA for the Data with Random decimal Point Positions”,2013 IEEE 9TH International Colloquium on Signal processing and its Applications,8_10Mac.2013,Kuala,Lumpur,Malaysia.

[6] SalvadorE.Tropea*,”*FPGA implementation of base-n logarithm**”,** Electrónicae Informática, Instituto Nacional de Tecnología Industrial Buenos Aires, Argentina

[7] P. K. Meher, j. Valls, t. -b. Juang, k. Sridharan, and k. Maharatna, “50 years of cordic: algorithms, architectures and applications,” *ieee transactions on circuits and systems i: regular papers*, vol. 56, no. 9, sep. 2009.

[8] A. M. Mansour, a. M. El-sawy, m. S. Aziz, and a. T. Sayed,”a new hardware implementation of base 2 logarithm for fpga”, *international journal of signal processing systems vol. 3, no. 2*, december 2011.

[9] J.A. Jensen: *Field: A Program for Simulating Ultrasound Systems*, 10th Nordic-Baltic Conference on Biomedical Imaging Published in Medical & Biological Engineering & Computing, pp. 351-353, Volume 34, Supplement 1, Part 1, 1996. M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.