

# FPGA Based Sound Location Estimation using the Grid Search Method

Sugavasi Mrudhula  
VLSI System Design  
SriSai Institute of science and Technology  
Rayachoti, Kadapa, Andhra Pradesh, India

Shaik Kaleem Basha  
Assistant Professor In Department Of Electronics And  
Communication  
Sri Sai Institute Of Science And Technology  
Andhra Pradesh , India

**Abstract**—Audio sound source Localization plays a important role in signal processing field. The modern Source Localization Technique uses signal processing with combination of powerful parallel signal processing capacities of Time Delay Of Arrival (TDOA) algorithm and algebraic cost function then sound source can be mapped directly within the area of interest rather than giving a particular sector. The time delays are easily computed using cross correlation function using Grid Search Method between the signals from BLOCK ROM or Microphones .This implementation will reduce the number of microphone modules being used and moreover it maps the sound source by giving its coordinate (x, y) . This system works on the FPGA Platform of SPARTEN 3E Family by building a SOPC system. It will be more sophisticate to many engineers.

**Keywords**—BLOCK ROM, TDOA, Cost function, Cross correlation, FPGA, Spartan 3E Family, SOPC.

## I. INTRODUCTION

Audio sound source detection system will be implemented with VHDL and the algorithm will be synthesized to Xilinx FPGA .the complete setup will be demonstrated with a consideration that three microphone received signals are generated within the FPGA and intentional delay is induced to the received signals. This is generated on FPGA itself because the environment in which the system is tested is noisier which confuses the system to which direction it has to point. on the model of geometric propagation of the signal in an isotropic and homogeneous medium. This means that, given a TDOA measurement between three receivers placed on the linear plane position. This technology has been widely used in the video conference, Monitoring system,

It is suitable for night time security of commercial centers like Location Estimation Algorithm is used to estimate the time delays Where the sound is located on the free space. The major blocks inside FPGA will be Input signals to Channel A, Channel B and Channel C, Delay Insertion, cross correlation and location estimation. Results will be demonstrated with consideration that the three channel inputs are identical to each other with added delay to the channel .However, all of them are based jewelry shops, automobiles, electronic goods, etc. sound source localization technique can also be used in multimedia applications and engineering practice.

## A. Existing method:

Using Real time audio sound signals by using microphones from any sound source. In that we are using converts, acoustic filters, Mux and DeMux. It looking very complex and expensive. Robotic Applications are require intelligence to sense the position of the particular sound source point.

## B. Proposed Method:

In this project a sound source localization system will be modeled and will be implemented with VHDL. The algorithm will be synthesized to Xilinx FPGA. The complete setup will be demonstrated with the Block RAM's. The system employs Block RAM's with different audio samples and calculates the direction of the sound source using TDOA (time-delay of arrivals). All the functions for the sound source localization are implemented with the use of algorithms for acoustic signal capturing, cross correlation, short-term energy and azimuth computation. Modelsim Xilinx Edition (MXE) and Xilinx ISE will be used simulation and synthesis respectively. The Xilinx Chipscope tool will be used to test the FPGA inside results while the logic running on FPGA. The Xilinx Spartan 3 Family FPGA development board will be used this project.

## II. SOUND SOURCE LOCATION ESTIMATION

Time delay estimates (TDE) based methods uses the fact that the sound reaches the microphones with slightly different times. The delays are easily computed using cross-correlation function between the signals from different microphones. Variations of this approach use different weighting (maximum likelihood, PHAT, etc.) to improve the reliability and stability of the results under noise and reverberation conditions.

As most of the microphone arrays today have more than two microphones there are several ways to compute the overall direction. Finding the direction from all possible pairs and averaging it doesn't work well in case of

reverberation. The most common method is testing the hypothesis for direction of arrival using the sum of all cross-correlation functions with proper delays.

### III. STATE OF ART

Delay generate based on coordinates: This module generates the delay for three sensors arrangement based on the delay. This becomes useful to test the application without having the sensors connected. The code accepts the location coordinates and generates the relative time delays (the minimum delay will be subtracted from all the three delays). Hence for one of the sensors the delay will be zero. The preprogrammed ROM shall generate the delay based on the following assumption. If the result in the hardware (with sensors) must match with these results then the sensors must be arranged in the same

#### Distance estimation using grid search method

In this technique we are finding the coordinates by grid search method i.e we are considering the area of 7m by 7m (this is decided by the microphone detecting range) we take all the points in the area -3.5m to +3.5m at a distance of 0.125m.

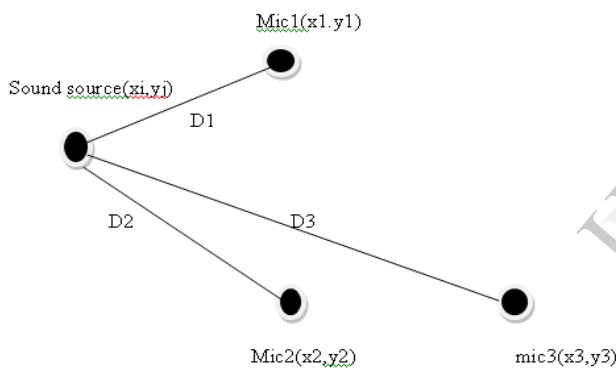


Fig. 1: Distance Estimation for Grid Search Method

The precision in finding coordinates of sound source differs, depending upon the distance at which we are taking the coordinates. We take whole grid coordinates by fixing "y" coordinate and incrementing the x coordinate ("x"=-32 to 31) till one iteration completes and then y coordinate is incremented like this the whole grid coordinates are taken and they are substituted in the below equations and

$$d1 = [(X_i - X_1)^2 + (Y_j - Y_1)^2]^{0.5}$$

$$d2 = [(X_i - X_2)^2 + (Y_j - Y_2)^2]^{0.5}$$

We find the difference between d1 and d2 we get the corresponding d2-d1,

$$Dis_{12} = T_{12} * \text{constant}$$

$$Dis_{23} = T_{23} * \text{constant}$$

$$Dis_{13} = T_{13} * \text{constant}$$

(Constant = velocity \* 1/f<sub>s</sub> = (330m/s)\*0.256msec (in Q11 format) (since f<sub>s</sub>=39062.5)

(T<sub>12</sub> we get it from cross correlation index, where as d1-d2 we get by assuming the coordinates of sound source in a specific grid)

$$\text{Diff}_{12} = (d2 - d1) - \text{dis}_{12}$$

Similarly

$$\text{Diff}_{23} = (d2 - d3) - \text{dis}_{23},$$

$$\text{Diff}_{13} = (d1 - d3) - \text{dis}_{13}$$

$$\text{Cost function} = (\text{diff}_{12} * \text{diff}_{12}) + (\text{diff}_{23} * \text{diff}_{23}) + (\text{diff}_{13} * \text{diff}_{13})$$

Where the cost function is minimum the corresponding x and y coordinates results the true sound source coordinates.

### IV. SOFTWARE DESIGN MODULE

#### A. Top level model:

The main objective of this project is to identify the sound source location. The top level diagram of the whole setup is shown above. From the ROM block we will get the audio samples and from that will from the 3 channels data as ch1\_data, ch2\_data and ch3\_data. The cross correlation module consists of 3 blocks namely cross correlation block 1 cross correlation block 2 and cross correlation block 3. The ch1\_data and ch3\_data would be fed to cross correlation block 1, ch1\_data and ch2\_data would be fed to cross correlation block 2 and ch2\_data and ch3\_data would be fed to cross correlation block 3. The cross correlation operation for the received signals would be done in their respective blocks.

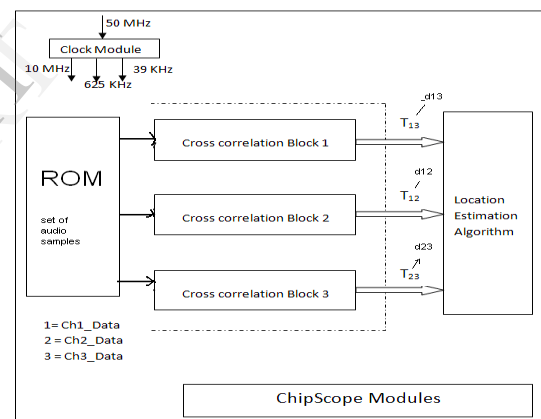


Fig. 2: Top Level Model Architecture

The output from the cross correlation blocks would be a time delayed signal. These time delayed signals T<sub>13</sub>, T<sub>12</sub> and T<sub>23</sub>. Location estimation block inputs are T<sub>13</sub>, T<sub>12</sub> and T<sub>23</sub>. Fine difference between inputs cross correlation and generates signal's which is used to find coordinates using the cost function and grid search method. Below it is explained in detail. By using real microphones we can capture real sound signal by using pc speakers or cell phones. Sound source from different places are generated accordingly to x and y coordinates of the sound source. second method is of storing the predefined values as there is the problem with balanced microphones

#### B. Cross correlation

We find out cross correlation by shift and multiply and not by FFT as it is a tedious method. Since the correlation is done on two signals at a time one signal will always keep shifting and the shifted signal values will be multiplied with the normal signal values at the respective time slot. The

output of the cross correlation block will be relative time delayed values of the received signal

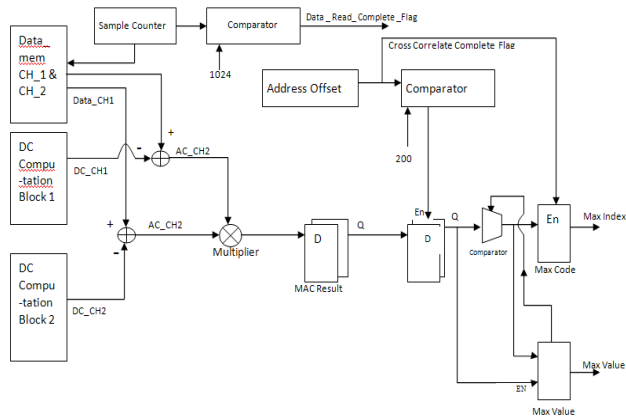


Fig. 3. Cross Correlation Model.

Two channels Channel A and Channel B outputs. Sample counter gives address for memory of data memory CH1 and CH2. The data from ‘Data Memory Channel A and Channel B’. The data from Data Memory Channel A is subtracted with the DC computation Channel A. Similarly for Channel B. The output of this subtracted values are ac\_ch1 and ac\_ch2. These two are performing a MAC (Multiply and accumulate). These two ac\_ch1 and ac\_ch2 are multiplied and the result is given to PIPO (Parallel Input and Parallel Output). Again the output of PIPO is fed back to accumulate. Apart from sample counter, one more counter ADDR\_OFFSET\_COUNTER which goes from 0 to 200. When ADDR\_OFFSET\_COUNTER counts to 200, then the cross correlation complete flag=’1’; which act as an enable for cross correlation value. When Sample counter counts to 1024, the data read complete flag=’1’. After completing of computing, the peak value resembles the direction.

C. Location estimation technique

There are two methods to take the input signal i.e., from the simulation (the two signals have equal magnitude and phase just they differ in delay, so by some sample delay for the signal, the different signals are produced) and the real time audio signal from the microphones. The microphones are placed in a right angle triangular fashion. They are connected to FPGA with matched (2M LONG) cables

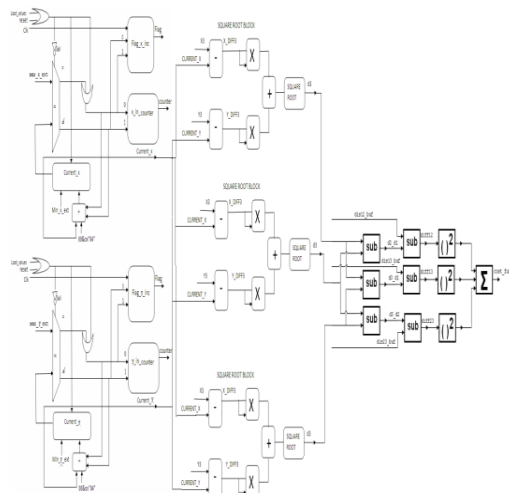


Fig .4:Location Estimation Module

D. Grid search method:

Most of the real world system models involve nonlinear optimization with complicated objective functions or constraints for which analytical solutions (solutions using quadratic programming, geometric programming, etc.) are not available. In such cases one of the possible solutions is the search algorithm in which, the objective function is first computed with a trial solution and then the solution is sequentially improved based on the corresponding objective function value till convergence. A generalized flowchart of the search algorithm in solving a nonlinear optimization with decision variable  $X_i$ , is presented in Fig.1.

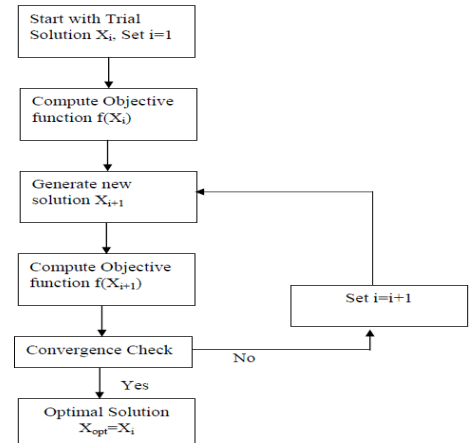


Fig. 5:Flowchart of Search Algorithm

This methodology involves setting up of grids in the decision space and evaluating the values of the objective function at each grid point. The point which corresponds to the best value of the objective function is considered to be the optimum solution. A major drawback of this methodology is that the number of grid points increases exponentially with the number of decision variables, which makes the method computationally costlier.

V. HARDWARE DESIGN

Chipscope

ChipScope is an embedded, software based logic analyzer. By inserting an “integrated controller core” (icon) and an “integrated logic analyzer” (ila) into your design and connecting them properly, you can monitor any or all of the signals in your design. ChipScope provides you with a convenient software based interface for controlling the “integrated logic analyzer,” including setting the triggering options and viewing the waveforms. Below Figure shows a block diagram of a ChipScope Pro system. Users can place the ICON, ILA, VIO, and ATC2 cores (collectively called the ChipScope Pro cores) into their design by generating the cores with the Core Generator and instantiating them into the HDL source code. We can also insert the ICON, ILA, and ATC2 cores directly into the synthesized design netlist using the Core Inserter tool. The design is then placed and routed using the ISE 9.2i implementation tools. Next, we download the

bitstream into the device under test and analyze the design with the Analyzer software..

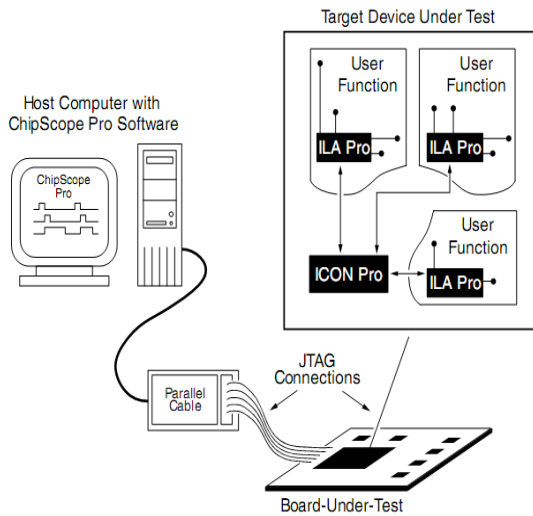


Fig. 6: ChipScope Pro Cores Description

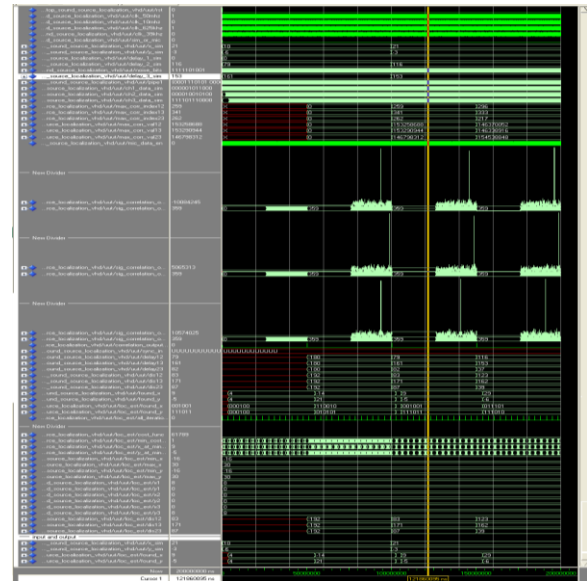


Fig. 7: Simulation Results

## VI. RESULTS

### A. Simulation results

Clk 50MHz clk coming from FPGA board.  
 Clk 10 MHz we divide 50/5 to get 10 MHz so that running logic for location estimation through delay at 10MHz frequency. running at high frequency as we gave to compare and estimate results for the whole grid. (that as to be done at the faster rate)  
 Clk 625 KHz—this is used by the ADC module i.e, 16 times higher than that of sampling frequency.  
 Clk 39 KHz – This is the sampling frequency used to capture the signal at high frequency rate (2820 kHz)  
 Sim or Mic – This is a control signal when it is active low then (i.e. '0') it is in simulation mode that is inbuilt x and y coordinates are given to process.  
 Sound source

### B. Chipscope results

After implementing the code on Xilinx FPGA, the results had been analyzed on ChipScope Pro Tool. Below are the results for various inputs and outputs

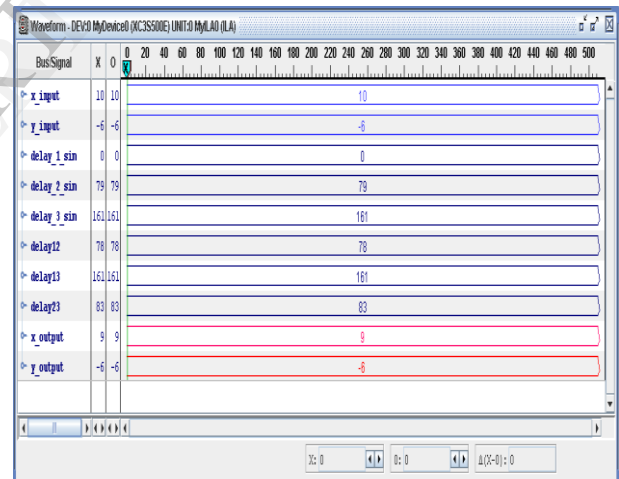


Fig. 8: Chip Scope Results

Results for the simulated inputs are shown below where the signals input\_x, input\_y are simulated inputs from where the sound signal is generated and signals output-x, output\_y (red in colour) are calculated values by the system. It can be observed that the simulated values and the calculated values are approximately same thereby verifying the correctness of the system.

### Future Scope

Although the technique implemented using just BRAM's brought a lot of improvements relative to previous techniques. Still there are some areas in which improvements can be made, such as:  
 The developed system is not precise for a noisy environment, if a technique is developed by which the

desired signal is analyzed instead of the maximum intensity signal then that system can be used in military applications. The available microphone set instead being same can't produce identical results, if some digital microphones are developed which can process the received sound signals and produce identical results then the precision can be increased. In the developed technique the calculations can be done for 3 decimal points because of which the results are not precise, if an algorithm is developed which can give precise results by using just 3 decimal values then the system becomes more accurate.

## VII. CONCLUSION

In this work we presented an audio signal source localization technique that is able to determine the coordinates (x, y) of sound source. The design is novel, cheap and simple. Active Strengths of this implementation are the simplicity of the design and the low cost, while a weakness is the exact coordinates can't be found but the approximate result can be found and this device works in

noise free environment. Application of this device is possible for security purpose.

## REFERENCES

- [1] S. Duy Nguyen, Parham Aarabi, Ali Sheikholeslami in their paper REAL-TIME SOUND LOCALIZATION USING FIELD-PROGRAMMABLE GATE ARRAYS discussed A real-time sound localization system was implemented on a Xilinx Virtex II 2000 (2V2000) FPGA.
- [2] Parham Aarabi, Alborz Mahdavi in their paper THE RELATION BETWEEN SPEECH SEGMENT SELECTIVITY AND SOURCE LOCALIZATION ACCURACY concluded The results in this paper illustrate the importance of NVTs for successful sound source localizations..
- [3] Mauricio Kugler, Kaname Iwasa, Victor Alberto Parciannelo Benso, Susumu Kuroyanagi, and Akira Iwata in their paper A COMPLETE HARDWARE IMPLEMENTATION OF AN INTEGRATED SOUND LOCALIZATION AND CLASSIFICATION SYSTEM BASED ON SPIKING NEURAL NETWORKS2001.

IJERT