3 Dimension Sound Source Localization with Cross-Correlation and CORDIC Algorithm on FPGA

Agung Nuza Dwiputra[1,a], Riko Hasiando Goknipasu Nainggolan[2,b] and Muhammad Arief Ma'ruf Nasution[3,c]
[1,2]Electrical Engineering Program Study, Institut Teknologi Bandung, Bandung. [a]Email: agungnuza@students.itb.ac.id,
[b]Email: rikostei2012@students.itb.ac.id, [c]Email: arief.maruf@students.itb.ac.id

***Abstract*- In this paper, brief explanation will be given about system that can localize sound source based on FPGA, this system use five microphones with predefined distance, which will be used to acquire the Time Delay of Arrival (TDOA) data. TDOA is the parameter needed by sound source to arrive at corresponding microphone. The focus in this paper is providing high accuracy 3D sound source localization with efficient area and energy. In this paper, the solutions are achieved by designing 3D sound source localization with efficient cross correlation process, by using only one adder and one multiplier, this paper also use CORDIC algorithm to create more efficient system. The design is then implemented using Altera Cyclone II achieving only 18237 logic elements and 8194 registers.**

***Keywords*—CORDIC,FPGA, Sound Source Localization, TDOA;**

I. INTRODUCTION

Sound source localization is a technology to detect source of the sound in dimensional space. Sound source localization (SSL) is already investigated for the past several years until today. There are many method to implement this system, one of them is by examining TDOA (Time Delay of Arrival). TDOA is usually applied in because it's a straightforward system. The problem is TDOA method need more high area and energy to achieve result that is more accurate. On the other hand, high accuracy detection is usually needed in its implementation. Thus, the improvement of energy and area are a challenging problem.

One of main process in TDOA method is cross correlation. Usually, GCC is applied in frequency domain as seen in [3]. Cross correlation process in frequency domain requires system to do FFT processing which consume more energy and area.

However, this paper provide method to directly apply cross correlation in discrete time domain to avoid FFT process. Thus giving more efficient FPGA area and power usage. Even though cross-correlation in time discrete domain still used a lot of multiplication and addition, this paper have provided better architecture to accommodate that problem. The cross correlation module architecture in this paper only used 1 adder and 1 multiplier, thus needing only small area and power consumption yet still produce high accuracy output.

II. BASIC THEORIES

A. Geometric Localization

Sound source localization system employs the time delay in determining the direction of the sound source. Many research have taking advantages of geometric position to determine time delay, such as in [4], [5] and [6]. In this paper, microphone array, which is being used, consists of five microphones arranged in three dimensions.

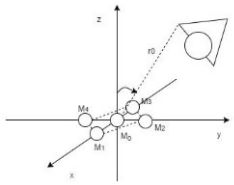

Figure 1. Five Microphone Geometry

If illustrated in a Cartesian coordinate, there will be five microphones which is represented each as M0, M1, M2, M3, and M4 sequentially that have the coordinate M0(0,0,0), M1$(L,0,0)$, M2$(0,L,0)$, M3$(-L,0,0)$,and M4$(0,-L,0)$. L represents distance between M0 microphone and the other microphone. L also represents the distance of microphone and the origin which is defined. If P represents the sound source point which has the coordinate (x,y,z) and r indicates distance between P and the coordinate origin, below equation will be valid.

$$x^2 + y^2 + z^2 = r^2 \quad (1)$$

The correlation between distance of the microphone, sound source, sound velocity, and time delay can be stated in this equation:

$$r = C\tau \quad (2)$$

(C represents sound velocity and τ represents time delay required by sound to go through some distance). From both correlation we can state distance of microphone 0 (M0) to another microphone (M1) with following equation:

$$r_0 - r_i = C(\tau_0 - \tau_i) = C\tau_i \quad (3)$$

Then, r defined as distance of sound source to the coordinate origin (M0) and τ_i represents time delay between microphone 0 and the other microphone (Mi).

Distance equation between microphone and sound source can be derived using each of microphone's coordinate with regard to equation (3).

The acquired equation will be processed in a spherical coordinate. Therefore, transformation process will be needed from Cartesian coordinate to spherical coordinate. After the coordinate transformation has been done, the notable parameters in geometric localization such as θ, φ, and r can be derived.

978-1-5090-3840-4/16/$31.00 ©2016 IEEE

$$r = \frac{(c^2 \sum_{i=1}^{4} \tau_i^2 - 4L^2)}{2c \sum_{i=1}^{4} \tau_i} \quad (4)$$

$$\tan\theta = \frac{(\tau_2 - \tau_4)[2r - c(\tau_2 + \tau_4)]}{(\tau_1 - \tau_3)[2r - c(\tau_1 + \tau_3)]} \quad (5)$$

$$sin\varphi = \frac{\sum_{i=1}^{4}\{[L^2 + 2rc\tau_i - c\tau_i^2]\cos[(\theta - (i-1)\frac{\pi}{2}]\}}{4rL} \quad (6)$$

B. *General Cross Correlation* (GCC)

Based on the amount of microphone being used in the form of array configuration with predefined distance, estimation about sound source can be completed by implementing general cross correlation process. Based on the cross correlation process that has been done, Time Delay of Arrival (TDOA) can be acquired and will be used to estimate the direction of sound source that enters the system.

Cross correlation between two signals from adjacent microphone which received by the system will give some TDOA values for different adjacent microphone combination. Selection for the adjacent microphone combination that gives the minimum value of TDOA will be done which will represent that the sound source location close to corresponding adjacent microphones. Computation of TDOA value can be done by modeling the sound source that microphone receives according to following equation as seen in [1].

$$x_i(t) = A_i s_i(t - \tau_{TDOA}) \quad (7)$$

A_i shows amplification of the sound source, τ_{TDOA} shows time delay until sound has been received by the corresponding microphone. For this system design initial condition has been given that the case is for only one sound source. Then Fast Fourier Transform (FFT) are implemented to the signal that each microphone acquired so representation of signal in frequency domain is limited to some ranges in order to facilitate acquired sound source processing. Frequency cut-off selection is based on the possibility range of sound source so processing that will be done can be minimized but all of the information about the sound source still appear, filtering also will be performed to eliminate noise. Based on the FFT process, sound source which has been received by microphone in frequency domain will be given as follows.

$$X_i(f) = A_i S_i(f) \quad (8)$$

Cross correlation process then applied to each possible microphone pair. Cross correlation process can be achieved using equation as follows.

$$R_{x_i x_j}(\tau) = \int_{\omega_{min}}^{\omega_{max}} X_i(\omega) X_j(\omega) e^{j\omega\tau} d\omega \quad (9)$$

From the cross correlation result, peak of each result will be checked to acquire TDOA for each microphone configuration as follows.

$$\tau_{TDOA} = \arg\max_{\tau} R_{x_i x_j}(\tau) \quad (10)$$

C. CORDIC Algorithm

CORDIC algorithm is used to calculate angle values that can determine the direction of sound source. The explanation about CORDIC algorithm will be given below.

- Arc tangent

Arc tangent calculation can directly be done by using vectoring mode from CORDIC rotator if the addition for each angle starts from zero values. Arc tangent computation in the designed system take benefit from the time delay that has been received from two microphones configuration. In general, following equation will be valid.

$$x_{i+1} = x_i - y_i d_i 2^{-i} \quad (11)$$

$$y_{i+1} = y_i + x_i d_i 2^{-i} \quad (12)$$

$$z_{i+1} = z_i - d_i tan^{-1}(2^{-i}) \quad (13)$$

With

$$d_i = +1 \; if \; y_i < 0. and - 1 \; for \; other \; value$$

In the last iteration, corresponding arc tangent value can be acquired as follows.

$$z_n = z_0 + tan^{-1}\left(\frac{y_0}{x_0}\right) \quad (14)$$

z_0 represents initial angle phase and $z_n$represents desired value by giving value for $x_0$and $y_0$in the beginning.

- Cosine

Cosine value is used to find the direction of sound source so another CORDIC algorithm should be used to compute the cosine value. In this system, cosine value will be searched with equation (11), (12), and (13) with some given requirements.

y_0 will be initialized with 0 value, that way the desired cosine value can be determined by using equation as follows.

$$x_n = A_n x_0 \cos z_0 \quad (15)$$

With provision that x_n represents the desired cosine value by first arrange the value of x_0 to $\frac{1}{A_n}$ and z_0 represents the desired angle value. [2]

III. HARDWARE DESIGN

In designing the sound detector using FPGA, signal that will be received by microphone as the input need to be processed first in order to make signal can be processed digitally. Signal acquired by microphone need to be amplified, filtered and complete ADC process.

For amplification process, acquired signal by microphone will be amplified. This process need to be done because the acquired signal are to low and hard to be processed. After the amplification process, filtering process then implemented. Environment and interface that appear in this system will give some noise to the received signal. Thus, input signal need to be processed to eliminate all of the noise. Since sound wave that human can hear appear in the range of 20 Hz to 20 KHz, band pass filter is needed to eliminate all of the noise. This band pass filter is created analogously.

Following the filtering process is the signal conversion from analog to digital. Processing in FPGA has to be done digitally, in order to fulfill that, the input signal need

to be converted to digital representation. ADC process has to be done to sampling and quantize the data.

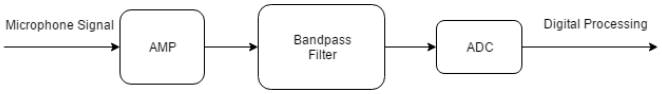

Figure 2. Analog signal conditioning process

Generally, analog signal conditioning process can be done using above figure. After analog processing, signal can be processed digitally. Digital processing can be done to acquire sound source position in three dimensions. The output from the system being used is sound source position in spherical coordinate form. In digital processing, after the signal converted with signal sampling will be passed to Hanning window to acquire output. Following that, the output in frequency domain will be used and cross correlation will be performed between each signal that has been received by microphone. Next process is to detect the peak from the cross correlation result so delay value of each microphone pair can be acquired. Delay from each microphone pair will be used to arithmetically calculate the geometry condition which in the end will give the estimation of the sound source position corresponding to the microphone. Block diagram for digital processing of the signal is given below.

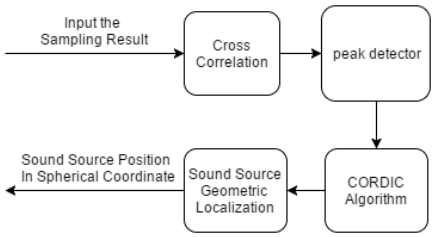

Figure 3. Digital signal processing process on FPGA

IV. SIMULATION RESULT

A. Cross Correlation Simulation

One of the notable subsystem in sound source detection system is cross correlation. Cross correlation can determine the time delay of each microphone which is essential to determine sound source position. Simulation has been done using MATLAB and the result can be seen on figure 4 below. The simulation is conducted by using 2 signals which are the reference signal and the delayed version of the reference signal. The second signal is delayed by 6 samples from the reference signal.

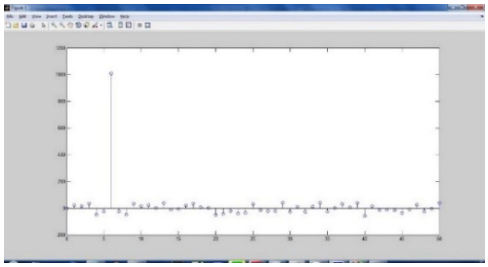

Figure 4. Cross correlation simulation on MATLAB

In determining time delay, maximum value is examined from the cross correlation result of microphone input signals. In this simulation, cross correlation successfully acquired the time delay with the correct value of six samples.

B. 3D Sound Source Position Calculation Simulation

To compute the position of sound source location, the previous derived equation is used. The derived equation then need to be verified. Verification performed by making the code for the corresponding equation in C language then compile and run the source code. The code requires 3 inputs which are the position of the sound source in Cartesian coordinate. The position is used to determine the travel time of sound from the sound source to each microphones. It is now possible to calculate the geometric location of the sound source by utilizing the time differences of arrival which are obtained from the travel times. From several attempts, the results are acquired as follows.

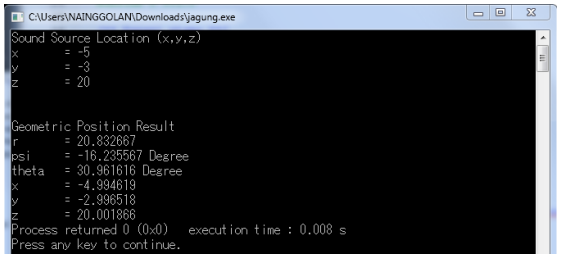

Figure 5. Calculation testing for position (-5,-3,20)

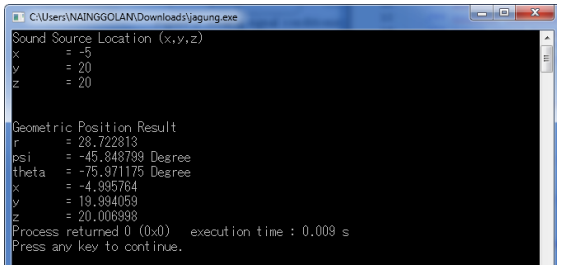

Figure 6. Calculation testing for position (-5,20,20)

From the acquired simulation result, it can be seen that the derived equation successfully calculate the sound source position accurately. This can be seen from the error value of the calculation, which gives a relatively small value between input position and position that has been calculated.

V. IMPLEMENTATION

A. Cross Correlation Module

Cross correlation in discrete time system can be written as equation (22).

$$R_{x_1x_2}(j) = \sum_n x_{1_n} x_{2_{n-j}} \quad (16)$$

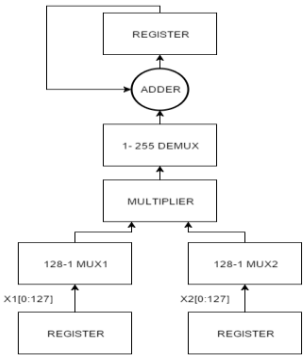

Figure 7. Cross-Correlation Module Design

Cross correlation calculation is similar to discrete convolution. It requires many multiplication and addition process. Cross correlation of 2 discrete signals with N samples each includes N^2 multiplication and addition processes. This computation consumes a large area. Therefore, we construct an architecture to compute GCC without using an excessive number of multiplier and adder. In fact, this architecture only utilizes one adder and one multiplier. The design used to implement equation (22) in hardware is shown in Figure 7.

The two input signals are stored in two 8-bit registers. 8-bit registers are used because the ADC output is an 8-bit digital signal. Each register is capable to store up to 128 samples. The sequence of data inside the registers are required to calculate the cross correlation and then ultimately determine the time difference of arrival. In the 3D localization simulation, the time differences of arrival between all microphones with the reference microphone are about -250 µs to 250 µs. Therefore, the sampling frequency and the clock needs to be set to50 kHz. This is necessary to ensure that the 128 samples signal are able to cover the maximum time delay of 250 µs. One sample is acquired every 2 µs, which means the delay of 127 samples is equal to the time delay of 254 µs.

We know that the product of each sample in both signals is required to calculate the cross correlation of the two signals. The cross correlation computation of 2 signals with 128 samples requires 16384 multipliers and adders which is an enormous number. Therefore, a different solution is essential for this matter. The 128to1 multiplexer in Figure 8 is the key to solve this problem. The multiplexer selects the samples that will be multiplied from both signals and then send the samples to the multiplier. This architecture solves the problem of area. However, more clock cycles are used to complete all multiplications.

The cross correlation output of two signals with 128 samples each will have 255 elements. It is essential to add the multiplication result to the right index of the output register. The 255to1 demux is used to determine the address of the correct element or index. Finally, the index that contains the largest element is detected by using peak detector. This index corresponds with the time difference of arrival between the two signals.

The design is verified by using two discrete signals. The signals are provided in Figure 8 and Figure 9.

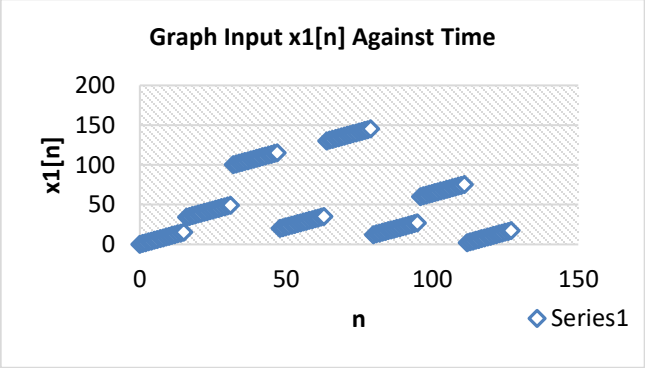

Figure 8. Graph x1[n] Against Time

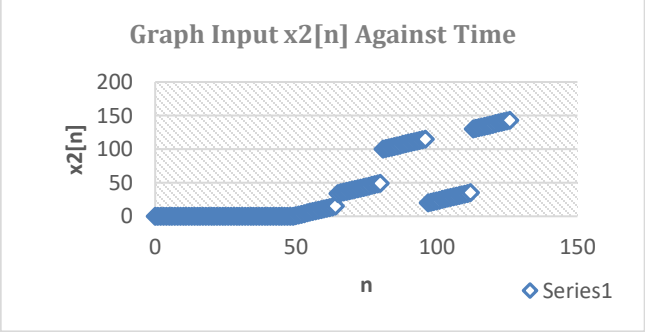

Figure 9. Graph x2[n] Against Time

The input signal x2[n] is the signal x1[n] delayed by 49 samples. The output waveform of cross correlation module can be viewed in Figure 10.

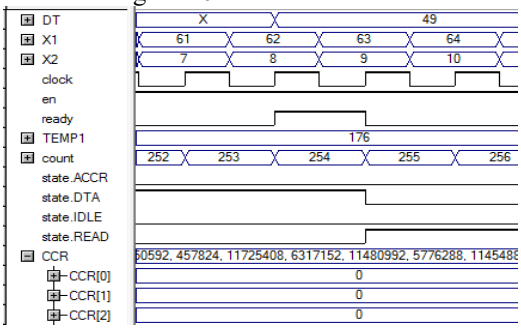

Figure 10. Functional Simulation Result of Cross correlation Module

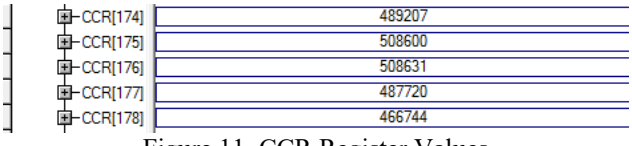

Figure 11. CCR Register Values

Output port DT indicates the sample delay between the two signals. The sample delay is 49 that matches with the delay given to the x2[n] signal. The CCR register in Figure 11 is the register that stores the cross correlation outputs. Figure

11 shows that the correlation result with the index 176 gives the largest cross correlation value. Thus, the sample delay that is detected by the cross correlation module can be calculated using equation (23). The calculation result is the value that is shown by port DT in Figure 10.

$$DT = Peak\ Crosscorrelation\ index - 127 = 176 - 127 = 49 \quad (17)$$

B. CORDIC Module

Based on the equation in subsection II.A, CORDIC module is required to calculate arctan and cosine. CORDIC algorithm to calculate these values is implemented by making an arctan table for angles, which generates the tangent value of 1, ½, ¼, etc. Those angles are first mapped to 32-bit numbers with equation (24) before they are stored in the table.

$$Mapping\ result = \frac{\theta \ll 32}{360} \quad (18)$$

The values stored in the arctan table are used to calculate the arctan and cosine of the input using iteration of the equations provided in the subsection II.C. The functional simulation result of CORDIC cosine module can be seen in Figure 12.

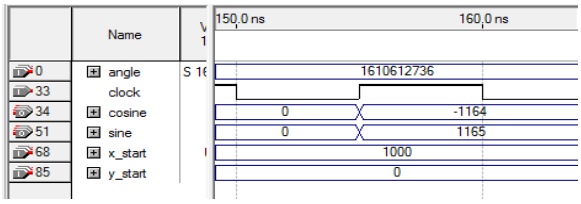

Figure 12. Functional Simulation Result of CORDIC Cosine

The number of iterations depends on the width of the input data. In this case, the input data is 16 bits wide, thus 16 clock cycles are required to complete the cosine calculation. The cosine result can be seen in Figure 12 at the 16th clock cycle. The CORDIC algorithm produces a gain of 1.647 and the x_start value is 1000. The angle input used in the simulation is 135°. This value is mapped into a 32-bit number. The exact angle value can be determined using equation (19).

$$\theta = (mapping\ result * 360) \gg 32 = 135° \quad (19)$$

The calculation that is conducted in this simulation is provided below. The output port cosine shows exactly the same result as manual computation.

$$x\_start \times gain \times \cos(angle) = 1647 \cos(135°) = -1164 \quad (20)$$

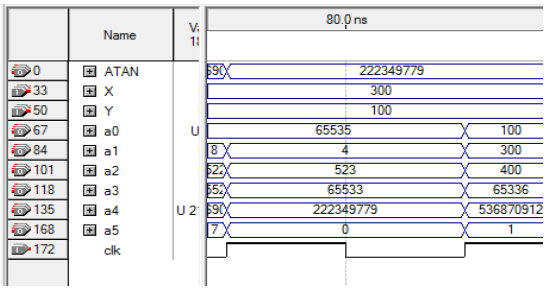

Figure 13. Functional Simulation Result of Arctan Module

The number of iterations required in this module depends on the size of the arctan table. The arctan table, which is used in this module, contains 8 angles, thus 8 clock cycles are required to obtain the arctan result. Figure 13 shows that the arctan module is given the input X 300 and Y 100. The equation that is calculated in this simulation is shown in equation (21).

$$\arctan\left(\frac{Y}{X}\right) = \arctan\left(\frac{100}{300}\right) = 18.43° \quad (21)$$

The output port ATAN indicates the angle value which has been mapped into a 32-bit number. The actual output of the module can be calculated using equation (22).

$$\theta = (mapping\ result * 360) \gg 32 = 18.64° \quad (22)$$

There is a slight difference between the output of the module in equation (22) and the actual arctan result in equation (21). This is caused by the number of angles stored in the arctan table. The more angles stored in the arctan table, the more accurate the approximation of CORDIC algorithm. However, the clock cycles required to complete the calculation will also increase.

C. Geometric Localization Module

This module receives the output of CORDIC and cross correlation module as inputs. These values will be processed and calculated according to equation (4), (5), and (6) in order to determine the value of r, θ, and φ, which corresponds to a certain location inside a spherical coordinate. Figure 14 shows the functional simulation result of this module.

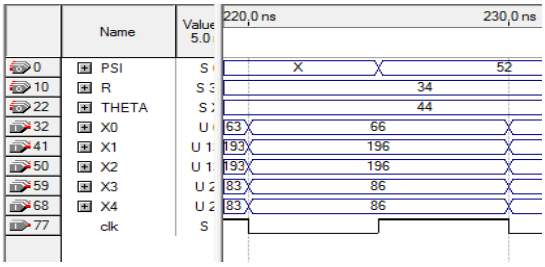

Figure 14. Functional Simulation Result of Geometric Localization Module

The simulation is done by giving the time delay in equation (23) as inputs.

$$td_1 = td_2 = 135\mu s;\ td_3 = td_4 = -192\mu s \quad (23)$$

These values are acquired by using the simulation in subsection IV.B. The time delay values will be obtained if the sound source is located at (20, 20, 20) in Cartesian coordinate system. The simulation result is shown in Figure 15. Therefore, the geometric localization result is expected to be same with the C simulation result.

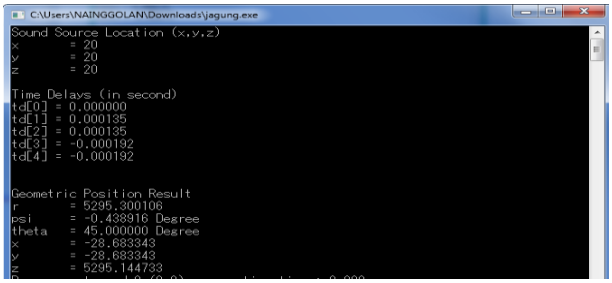

Figure 15. Geometric Localization Simulation using C Language

The value of r, θ, and φ in the waveform and the C simulation result are nearly the same. The module approximates the coordinate in integers, which is the reason why it is not as accurate as the C simulation. However, the module is capable of calculating the geometric location of sound source based on time difference of arrival.

D. Benchmarks

The benchmarks shown in Table I compare the number of logic elements used in the proposed design and a reference paper. The benchmarks are acquired using the same method and number of inputs. Thus, it concludes that the proposed design is able to produce an accurate result by utilizing much less area than the other design.

TABLE I. BENCMARKS

Design	Logic Elements	Method
Paper [3]	1,192,793	GCC
Proposed	18,237	GCC

VI. SYSTEM IMPLEMENTATION

The system uses an array of five microphones to acquire the input signal. Each microphone circuit is designed using the schematic diagram in Figure 16.

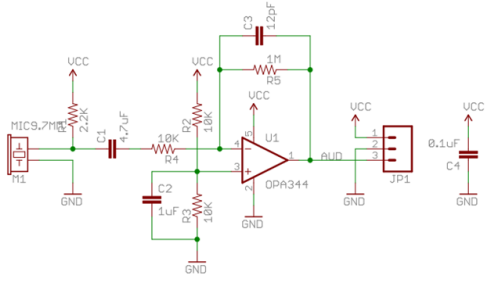

Figure 16. Schematic Diagram of Microphone Circuit

The gain of this circuit can be adjusted using equation (24).

$$gain = \frac{R_5}{R_4} \quad (24)$$

The gain is set to 40dB, which is equivalent to the amplification of 100. The circuit filter for this system is set as band pass from 3,3 Hz up to 13,2 KHz. Finally, IC ADC0820 is used to convert the analog signal to digital signal that can be processed.

The system uses cross correlation to determine time difference of arrival, thus we configured an array of microphones according to the configuration in Figure 1. Each microphone is located 10 cm away from the reference microphone. The array of microphone and circuit used are shown in Figure 17.

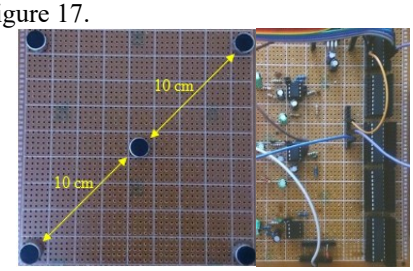

Figure 17. Array of Microphones

The verification of each microphone's functionality includes the observation of ADC0820 IC output signal with a sampling rate of 50 kHz. The change of bit value occurs as the microphone obtains the sound signal. The outputs of the ADC are shown in Figure 18.

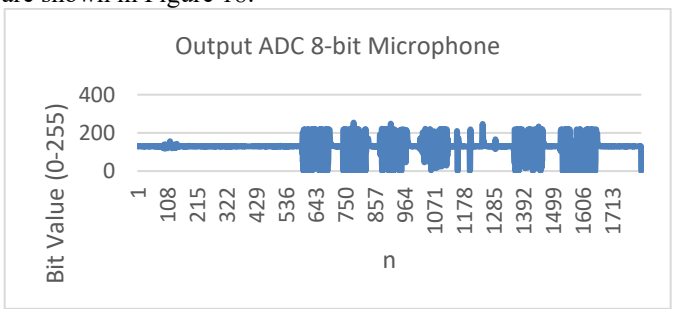

Figure 18. ADC Output

VII. CONCLUSION

Sound source localization are implemented using efficient cross correlation in discrete time domain, thus reducing area and energy for FTT process. This system is using 5 microphone as input for calculating TDOA process. The system are capable to reduce area and energy while maintaining high accuracy in the system. Overall simulation on FPGA show that sytem only need 18,237 logic elements and 8194 registers.

REFERENCES

[1] Hu, Jwu-Sheng , et. all , *Sound Source Localization by Microphone Array on a Mobile Robot Using Eigen-Structure Based Generalized Cross Correlation,* IEEE Paper, 2008, pp. 1-6.

[2] Meher, Pramod K., et. all, *50Yearsof CORDIC : Algorithms, Architectures and Application,* IEEE Paper, 2009, pp.1-12.

[3] Nguyen, Duy dkk. *Real-Time Sound Localization Using Field-Programmable GateArrays,*IEEE Journal, 2003.

[4] Alameda-Pineda, Xavier and Radu Horaud. *Geometric Approach to Sound Source Localization from Time-Delay Estimates*, IEEE Journal, 2014.

[5] Li, Jinghong dkk. *Design and Implementation of Sound Source Localization System Based on FPGA*, IEEE Journal, 2012.

[6] Kan, Yue dkk. *Passive Acoustic Source Localization at a Low Sampling Rate Based on a Five-Element Cross Microphone Array*, MDPI Journal, 2015