DAT096
Half-Time Report

Group M3

Ashwin Kumar Balakrishnan
John Croft
Prema Manickavasagam
Hezhe Xiao

March, 2020

CHALMERS
UNIVERSITY OF TECHNOLOGY

Abstract

Abstract goes here eventually.

Contents

1 Introduction **1**
- 1.1 Problem description . . . 1
- 1.2 System Requirement Specifications . . . 1
- 1.3 Scope . . . 2

2 Background Theory & Prior Work **3**
- 2.1 Beamforming . . . 3
  - 2.1.1 Sensor Arrays . . . 3
  - 2.1.2 Delay-and-Sum (DAS) . . . 5
  - 2.1.3 Mathematical Background . . . 6
- 2.2 Audio Location . . . 6

3 Hardware Design **7**
- 3.1 Xilinx FPGA Artix-7 AC701 . . . 7
  - 3.1.1 FPGA: inherent advantages and disadvantages . . . 7
- 3.2 Microphone Array and interconnect . . . 8
- 3.3 Breakout board . . . 9

4 Implementation **10**
- 4.1 System Architecture . . . 10
- 4.2 Design Considerations . . . 10
  - 4.2.1 Design multiple channels for CIC filter using the streaming interface . . . 10
  - 4.2.2 PDM to PCM conversion . . . 11
  - 4.2.3 DAS beamforming . . . 12
- 4.3 Data storage and communication (to PC) . . . 14
- 4.4 Reference Design . . . 14
  - 4.4.1 GRLIB and APB BUS . . . 14

5 Experiments **16**
- 5.1 Early experiments . . . 16
  - 5.1.1 Establishing functionality of multiple parallel/interleaved sensors using time-multiplexing . . . 16
- 5.2 Future Experiments . . . 16

6 Results **17**

7 Discussion & Analysis **18**

8 Conclusion & Future Work **19**

9 References & Acknowledgement **20**

Glossary/Abbreviations

GRLIB - Gaisler Research Library

MEMS - Micro Electro Mechanical systems

PCM - Pulse Code Modulation

PDM - Pulse Density modulation

SONAR - Sound Navigation and Ranging

DAS - Delay-and-Sum

CHAPTER 1 | Introduction

The main aim of this project is to locate an ultrasonic acoustic source in two dimensional space using a linear array of MEMS microphone sensors. This is done using a technique called beamforming, used to achieve directionality in signal transmission and reception.

The most common application that comes to mind when talking about ultrasonic source localization is how bats use ultrasonic sound waves to locate objects in space. This concept was adapted in SONAR, used by submarines to navigate underwater and communicate by determining the direction and distance of the transmitter.

The problem in this project is similar, but does not use echolocation or radar (it is strictly source to receiver) and does not detect distance.

1.1 Problem description

In this project, we have to design a system that tracks the location of a robot mounted with an acoustic source in two dimensional space. The beamformer is basically a spatial filter that boosts the signal from the look-direction while attenuating signals from all other directions. The reference design given converts the raw PDM data from the microphones to PCM data on a AC701 ARTIX-7 FPGA board. The data is then sent via the Ethernet to a computer by using GRMON. The system to be designed should operate in real time and direction of the beam should be changed dynamically to locate the source.

1.2 System Requirement Specifications

The system to be designed should not contain any movable parts and should be able to :

- track the source in two dimenstional space.
- detect change in position of the source with a latency of less than one second.
- detect the source movements of more than two degrees within the cone of interest.

- use a fixed-frequency audio source.
- display the real time co-ordinates of the source on a PC.

1.3 Scope

The ultrasonic sound localization technique has a wide variety of applications. Apart from the ones mentioned above, it can be used to detect leaks in pressuried air sytems, water leaks in pipelines, directional speech enhancement. With industries becoming more automated nowadays, this can be also used to track robots. In this project, the goal is to implement beamforming using the delay-and-sum technique first and foremost. If that is successful and there is a reasonable amount of time left before the final evaluation, other techiniques may be taken into consideration.

CHAPTER 2 | Background Theory & Prior Work

2.1 Beamforming

Beamforming is a type of spatial filtering in which a signal receiver is more sensitive to signals from certain directions, while attenuating signals from other directions. It uses an array of smaller omnidirectional sensors to, in essence, emulate a larger, directional sensor.

This is in contrast to an omnidirectional receiver which is equally sensitive to signals from all directions, and hence performs no spatial filtering whatsoever.

The basic principle of operation is that the receiver is configured in some way (in terms of DSP operations and physical geometry of the sensor) to constructively interfere signals from the desired direction while destructively interfering signals from all others.

By configuring the DSP part of the receiver, the direction of sensitivity can be steered without physically moving the sensor array.

In one dimension, the sensitivity can conveniently be represented in a polar plot showing gain versus incident angle ψ. This is referred to as a **beam pattern** and an example can be seen in fig. 2.1.

The beam in the direction of sensitivity is referred to as the mainlobe. There are usually prominent, but attenuated, side-lobes on either side and, in the case of aliasing (discussed below), so-called grating-lobes which are unattenuated.

2.1.1 Sensor Arrays

Sensor arrays are receivers with multiple sensors arranged in a specific way.

The most basic type of sensor array is a equidistant linear array, in which sensor elements are positions along an axis with equidistant spacing between them.

In this type of sensor array, an incident wavefront will reach each sensor element at a different time, resulting in a phase shift on each. This is illustrated in fig. 2.2. A linear array can only direct its sensitivity in one dimension.

By combining multiple linear arrays, placed perpendicularly to each other, it is possible to extend directionality to all three spatial dimensions.

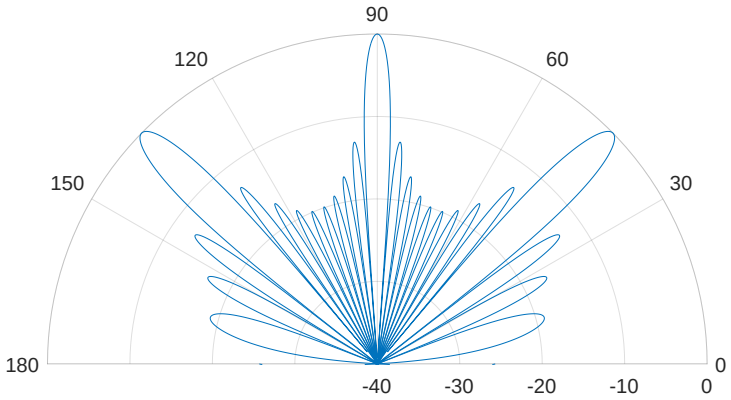

Figure 2.1: Beampattern illustration showing mainlobe, sidelobes and two grating lobes.

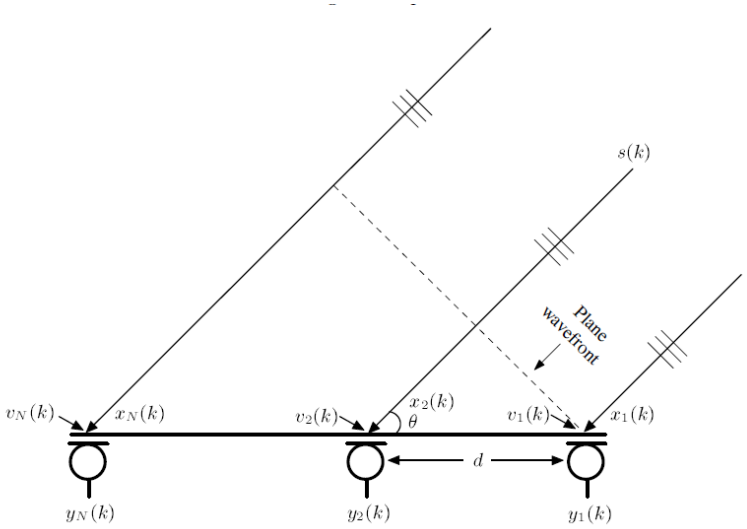

Figure 2.2: Pretty image illustrating wavefront, delays and constructive and destructive interference after wave summation. Nicked from J. Benesty and will be replaced by our own artwork at some point.

Other types of sensor array also exist, in particular composite arrays which are a simple way of partitioning a source signal containing multiple frequencies (broadband) into sub-bands that can be filtered individually[2].

Likewise, several algorithms exist for the DSP part of the beamforming receiver. A simple, but naive, approach is delay-and-sum (DAS). This approach is the one adopted for this project. DAS is naive in the sense that it does not use the values it produces.

More adaptive approaches, for example using a least-squares error minimization technique, use the values produced by the receiver as feedback to try to optimize some particular variable, such as signal strength, by steering the directional sensitivity. Ostensibly superiour in terms of system performance, such algorithms come at a cost of significantly more difficult theoretical and implementation challenges.

2.1.2 Delay-and-Sum (DAS)

Delay-and-Sum is a technique for steering the directional sensitivity of a sensor array. As previously discussed, an incident wavefront will reach each individual sensor at a different time (provided the signal does not originate perpendicular to the array axis, in which case all sensors receive the wavefront at once). In order to steer the directional sensitivity, the signal that each sensor receives is delayed by an amount of time corresponding to the desired direction. By doing this, the physical delay between sensors for a given incident angle can be compensated for. The delayed signal samples are then summed to give either constructive or destructive interference. If the delays match the incident angle (ie. the sensor array is configured to "look" in that direction), each sensor will delay its received signal such that they are all in phase and maximum value is obtained when they are constructively summed.

Weights may also be used in conjunction with delays shift, affecting the beam pattern, though this project will not focus on that aspect.

Spatial Aliasing

As previously mentioned, the spatial separation of sensor elements in a particular sensor array results in a relative phase shift in the received wavefront in each sensor. When the incident angle and spatial separation are such that the phase shift is one wavelength λ or a multiple thereof, the direction becomes ambiguous. This is shown in fig. 2.3.

Aliasing is obvious in a beam pattern as the lobes are unattenuated.

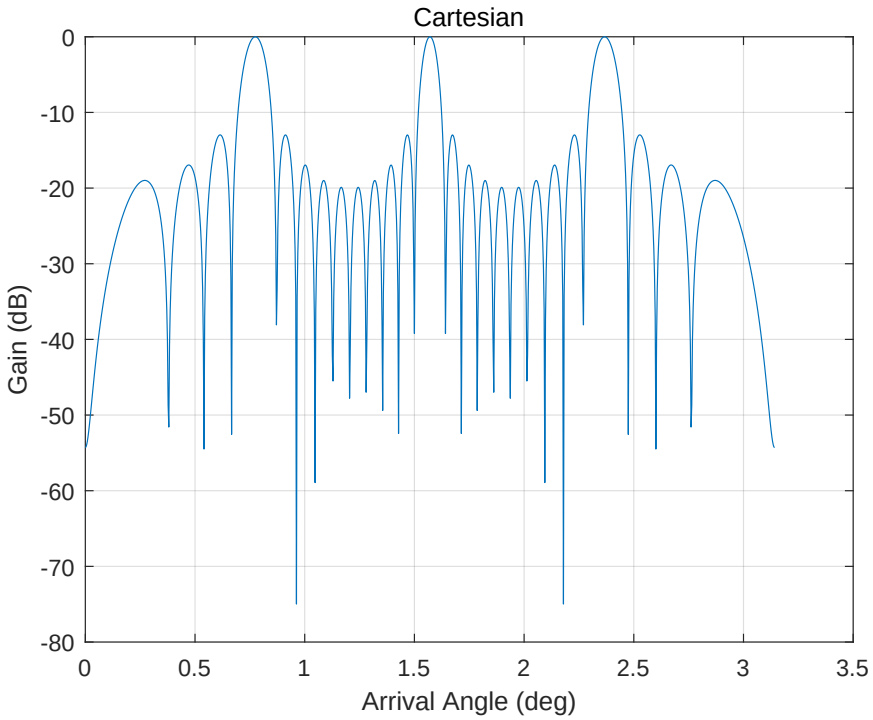

Figure 2.3: Image showing wavefront reaching sensors with same phase, and the resulting constructive interference giving aliasing.

2.1.3 Mathematical Background

2.2 Audio Location

The energy of the signals from the microphone are calculated using the following formula:

$$E_c = \int_{-\infty}^{\infty} \mid x_c(t) \mid^2 \; dt$$

where $x_c(t)$ is a continuous signal. For a discrete signal

$$E_d = \sum_{-\infty}^{\infty} |x[n]|^2$$

This is calculated for every angle and the one which has the highest energy value will be the direction of arrival. After the full sweep across the entire range the values from two microphone arrays placed perpendicular to each other are multiplied to get the highest energy signal and locate the source.[3]

CHAPTER 3

Hardware Design

3.1 Xilinx FPGA Artix-7 AC701

FPGA (Field-Programmable Gate Array)(fig 3.1) is a semiconductor integrated circuit consisting of configurable logic blocks and interconnects[6]. The logic blocks consist of components like flip-flops and lookup tables (LUT) as well as specialised DSP (Digital Signal Processing) circuits such as multiply-accumulators. The application range for FPGA is very wide because of its flexibility. They are programmed by hardware description languages to achieve the desired function. FPGAs operate in parallel and hence desirable to carry out various functions

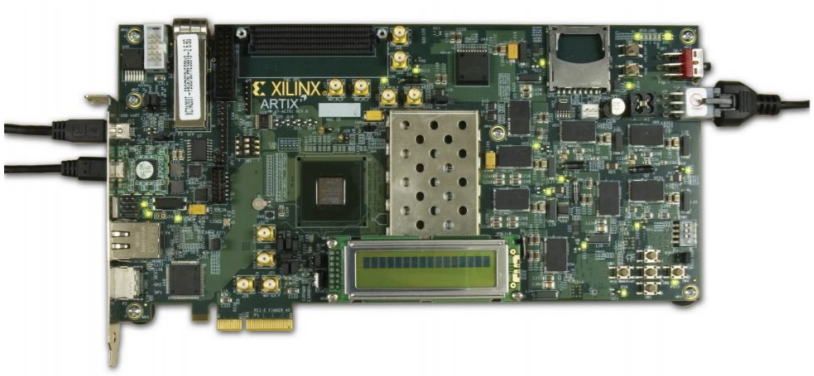

Figure 3.1: FPGA Artix-7 AC701

independant of each other without having to share the same resource at the same time.[7] The hardware platform used in this project is the Xilinx AC701 Evaluation Kit, using an XC7A200T FPGA and various peripheral devices such as DDR3 RAM. Xilinx Vivado is the main EDA (Electronic Design Automation) tool used for design and synthesis.

3.1.1 FPGA: inherent advantages and disadvantages

The inherent parallel computing of FPGA's make it faster and more efficient than other serial processors where the computation speed is low when compared

to the former. It is flexible because of the logic blocks which can be programmed to carry out the desired function. FPGA's can be reprogrammed easily without the need to rewire and the functionality can be changed faster. This feature also enables downloading algorithms and change them just like a computer changes program.[8]They are reusable and hence is very cost efficient and makes them ideal for prototyping. It eases up the product development and takes less time to market. They can be implemented in various real time systems because of its efficient architecture and high computation speed.[10]

FPGAs have a comparatively expensive unit price and designing for them is inarguably more cumbersome than a purely software approach. Some FPGA devices include processor cores on-chip, enabling a traditional software envionment (such as a Linux based operating system) to interface with the reconfigurable FPGA fabric. This approach tries to leverage the ease of programmability of an established hard-core processor (such as an ARM core) along with the extreme design flexibility of the FPGA. The FPGA used in this project does not have a built-in processor, though there exists Xilinx IP modules (Microblaze) to instantiate "soft" cores, processors that are constructed from the FPGA fabric directly. These are generally slower and consume area depending on their configuration, but otherwise fulfil the same purpose.

Furthermore, FPGAs have many IO connections (the XC7A200T-2FBG676C FPGA onboard the AC701 Evaluation Kit has 400 such pins), making them ideal for applications requiring many peripheral sensors and devices.

3.2 Microphone Array and interconnect

The microphone array(fig 3.2) used in this project is designed by syntronic and it consists of a linear array of 24 MEMS microphones.The distance between the microphones play a vital role in determining the frequency it can be optimized for. In the array used in this project the microphones are spaced 11.43mm

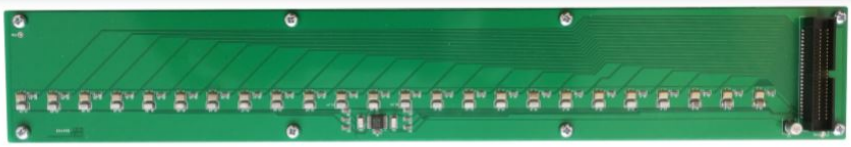

Figure 3.2: Microphone array

apart. The microphone used is SPH0641LU4H-1, which can be clocked at different frequencies to get different bandwidth.It is a miniature high performance microphone with one bit PDM output. This has to be filtered inorder to get a PCM output. It has RF immunity, has a high SNR of 64.3dB and supports ultrasonic applications.[9]

3.3 Breakout board

The breakout board (fig 3.3)is designed by Syntronic to act as a bridge between the FPGA and the sensor board. It sends the clock to all the microphones. It has 4 panels each which could accomodate 24 microphones each.

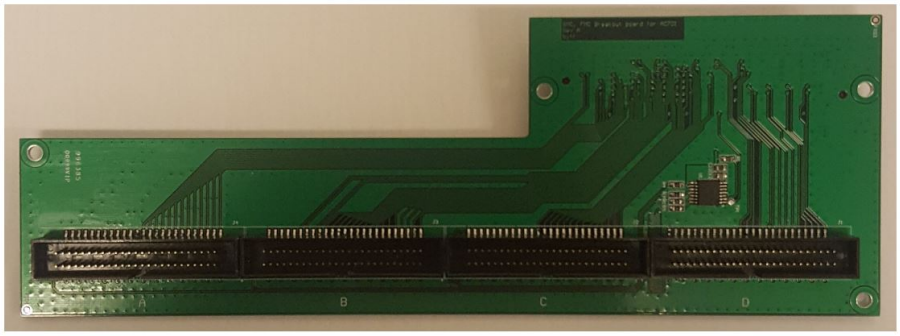

Figure 3.3: FMC Breakout board

CHAPTER 4 Implementation

4.1 System Architecture

The system architecture contains the process to convert the data acquired in 1 bit format from the microphone to PCM(pulse code modulation) in 16-bit format, delay-and-sum beamforming and transfer interface.The figure 4.1 of system architecture is shown as following.

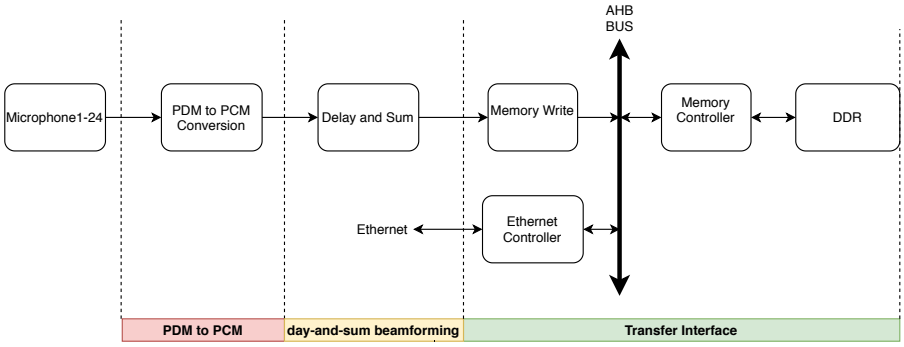

Figure 4.1: System architecture diagram

4.2 Design Considerations

4.2.1 Design multiple channels for CIC filter using the streaming interface

PDM data from multiple channels is input into CIC filter by Data Input Channel and processed by the Data Output Channel. From CIC filter schematic diagram 4.2, there are two channels, the channel master and the channel slave, to control CIC Compiler core to accept data. TVALID is driven by the channel master to show that it has data to transfer and TREADY is given by the channel slave to show that it is ready to accept data. The transfer happens until both TVALID and TREADY are high[5].

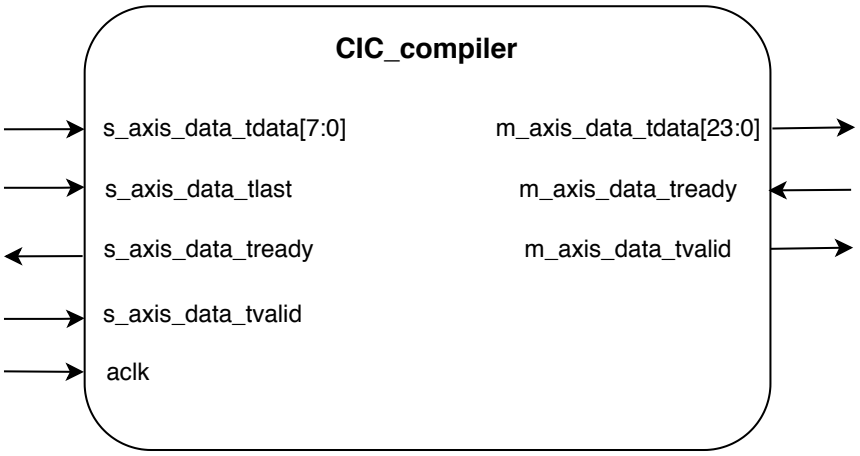

Figure 4.2: CIC filter schematic symbol

TDATA is the Data Input Channel and carries the unprocessed sample data. TLAST is also for the Data Input Channel and it is only used in multichannel mode, asserted by the sample data corresponding to the last channel. All of these use the AXI4-Stream protocol[5]. Figure 4.3 of time diagram shows this protocol of CIC filter input with four channels as an example. The design of multiple channels IP block mainly depends on it.

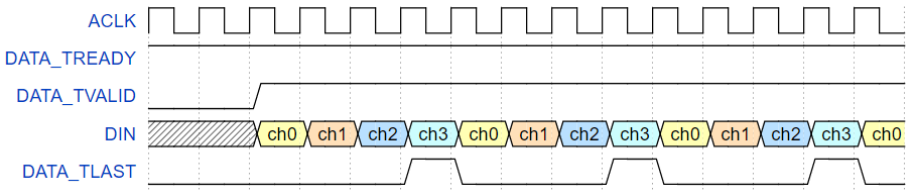

Figure 4.3: time diagram of CIC filter input

4.2.2 PDM to PCM conversion

In order to implement this conversion from PDM to PCM, the design uses one cascaded integrator-comb decimating filter(CIC filter), one half-band FIR filter and one low-pass FIR filter to output the 16-bit filtered data at a 44.1 kHz rate.

As the figure 4.4 shown, CIC Filter IP block is used to decimate the PDM input by a factor of 16 and to convert it into a PCM format, Halfband FIR Filter IP block is to decrease the sample rate of CIC output by 2 and compensate for the passsband of CIC filter which is not flat[3]. The last Low Pass FIR Filter IP block is to pass the low frequency and remove the high frequency noise.

Since the input sampling rate of filters depends on the input clock of the microphone system, the design has to generate appropriate clock inside FPGA. The PDM rate is 1.4112 MHz(44.1 Khz*32) based on the PCM sampling rate of 44.1

KHz of this design but because of the hardware limitations, it cannot generate 1.4112 MHz clock directly from 100 MHz as the input clock from Microphone system. Therefore, there are clock dividers used to guarantee the clock frequency down to 1.4112 MHz. Instead, a sample clock of 5.645 MHz is generated and divided down to 1.41125 MHz. The sample clock is therefore not exact, and it is as of yet unclear if this type of problem will cause problems with the system in future.

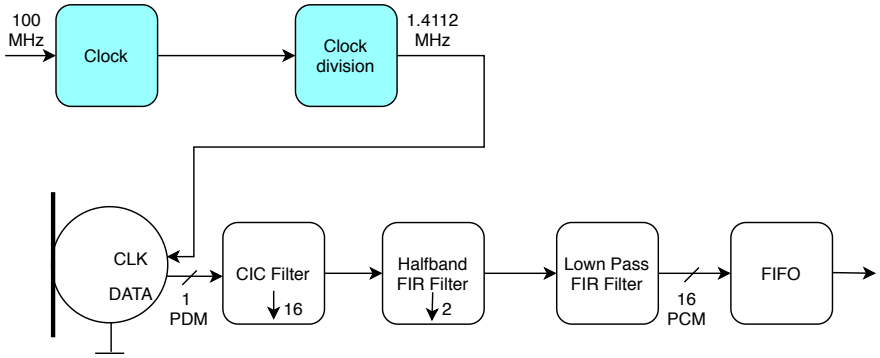

Figure 4.4: PDM to PCM conversion diagram

4.2.3 DAS beamforming

The delay values for each microphone with angles from 0 degrees to 180 degrees are pre-calculated in MATLAB. Assuming the source is in the far field and sound speed $\boldsymbol{c}$ is 343 m/s, $\boldsymbol{d}$ is the space distance between two neighboring sensors. The souce wave arrives one microphone with an angle of $\boldsymbol{\theta}$. The delay values between the $\boldsymbol{n}$th microphone and the reference sensors is decribed as flowing fomula[4].

$$\text{Delay} = \frac{(N-1)\,d}{c}\cos\theta$$

Using the calculation of $N_{clk} = ceil\,(Delay * f_{clk})$, f_{clk} is the frequency of clock of 100 MHz, to convert these daly values to a number of clock cycles and saving them in a COE file as the initialization of ROMs IP block in Vivado.

In the next experiment step, the clock frequency will be divided to drive the shift register IP blocks and using an adder IP block to add the maximum delay with the shift registers traversed by enable signals of FIR filters.

According to the formula below to get the plot of beam pattern with 24 microphones and sound frequency. $\boldsymbol{W}$ is the array's reponse to a signal, $\boldsymbol{f}$ is the sound frequency and $\boldsymbol{\Psi}$ is a directional angle from 0 to 180 degree (the incident angle of the wavefront) and $\boldsymbol{\theta}$ is the *look-direction* of the beamformer[4].

$$|W(\psi,\theta)| = \left|\frac{\sin\left(N\pi\frac{fd(\cos\psi-\cos\theta)}{c}\right)}{N\sin\left(\pi\frac{\mathrm{fd}(\cos\psi-\cos\theta)}{c}\right)}\right| \tag{4.1}$$

The array beam pattern can be seen in MATLAB with different number of microphones and value of source frequency, under the premise of fixed distance between two neighboring sensors, in order to help us analyse how the beamformer infected by the spatial aliasing. As we known, if the sidelobes is as low as possible, the signals from other directions except the source direction would be attenuated as much as possible so that we can determine the range of source sound frequency, it is about from 25 KHz to 30 KHz to avoid spatial aliasing.

Figure 4.5 plots the beam patter for 24 microphones with $\theta = \pi/2$ and f = 27 KHz. The mainlobe is properly at $\pi/2$.

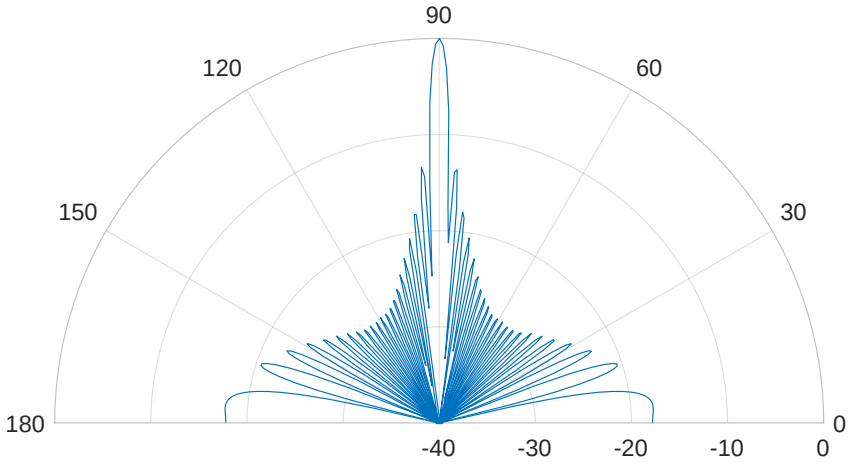

Figure 4.5: Beam pattern of 24 microphones with source frequency of 27 KHz

However, changing the *look direction* will adjust the angle of the mainlobe but also introduce new aliasing issues into the beam pattern.

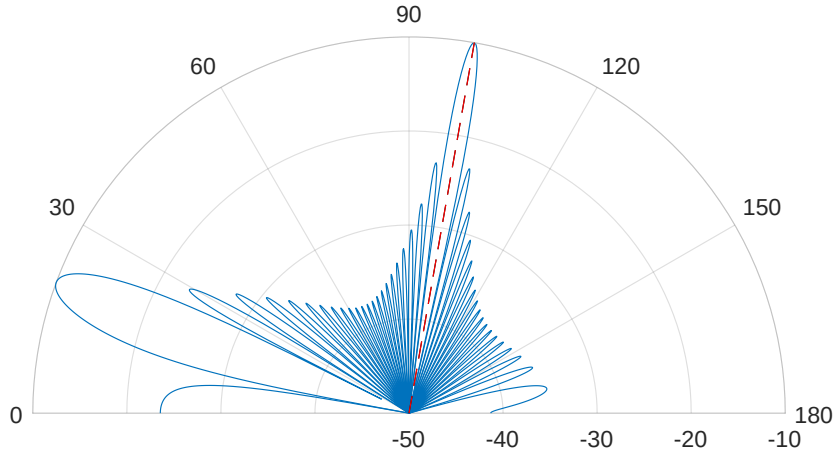

Figure 4.6: Beam pattern with f=27 kHz, d=11.43 mm and look direction at $\theta = 100°$. This results in aliasing

4.3 Data storage and communication (to PC)

As the system archtecture shown above, the PCM samples are saved in DDR on the board and read them from PC by Ethernet.

Therefore the memory controller, Ethernet controller and AHB bus are used to connect to FIFO IP block.

4.4 Reference Design

4.4.1 GRLIB and APB BUS

GRLIB is Gaisler Research Library to use for set reusable IP cores fro system-on-chip development using Gaisler's LEON3 processor, which is provided in the reference design.

It includes VHDL code for processor core, peripherals, bus and memory controllers and debug support unit to simulation and synthesis. LEON3 system is shown as following figure 4.6.

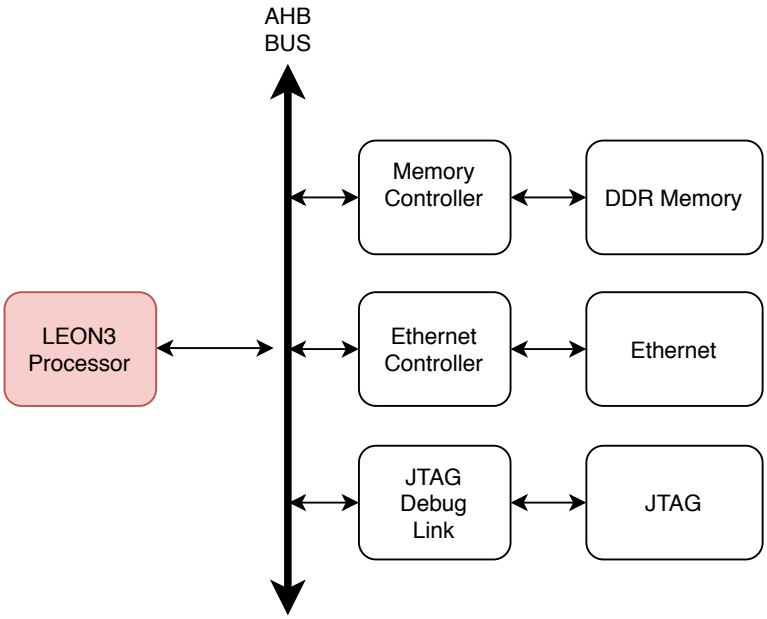

Figure 4.7: LEON3 based system

AHB Bus is AMBA High-performance Bus, which is multi-master bus for connecting peripherals with high data rates and variable latency. The structure is shown as figure 4.7.

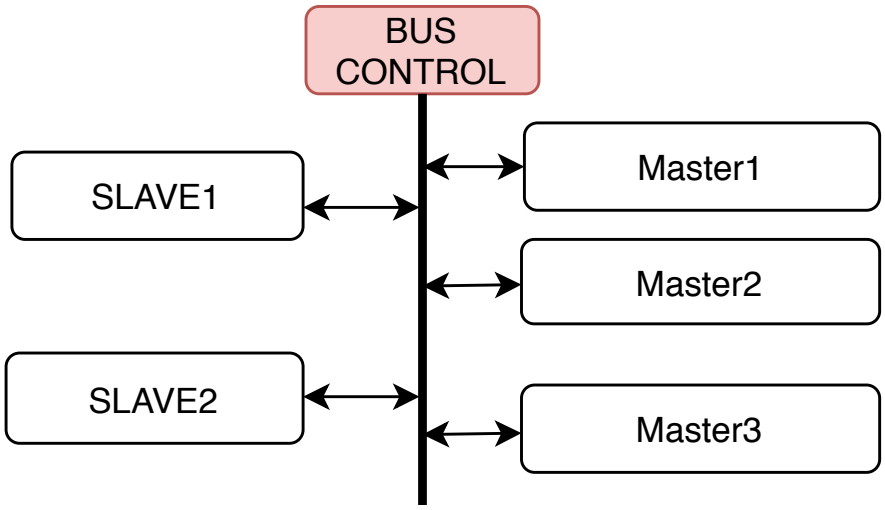

Figure 4.8: The structure of AHB Bus

CHAPTER 5 Experiments

5.1 Early experiments

5.1.1 Establishing functionality of multiple parallel/interleaved sensors using time-multiplexing

Verification of input signals in the audible spectrum?? (it could potentially also be used outside of the audible spectrum...) using Audacity.

5.2 Future Experiments

Implementing beamforming (using shift registers etc) and:

- Manual verification of sound source at fixed angle using fixed delay-values.

  Change angle manually using buttons.(get respective pre-computed delays from a source file for each increment/decrement in angle)

  Potentially use LCD for debug information (angle of beam etc...)

- Sweeping along one dimension at fixed rate

  What factors will limit the sweep rate/refresh rate? Memory

CHAPTER 6 | Results

The following results were obtained :

- The delay values for each microphone for every angle within the range of 0 degrees to 180 degrees is calculated by using the MATLAB script.
- By using the multi-channel mode in the CIC and FIR filter blocks given in the reference design, audio was recorded with multiple microphones. We listened to the recorded audio using AUDACITY and confirmed that the signals were sampled properly.

CHAPTER 7 | Discussion & Analysis

Even though we get the delay values for all angles, spatial aliasing occurs at the extremes i.e, 0 and 180 degrees. But with increasing number of microphones, spatial aliasing decreases as shown in figure 7.1.

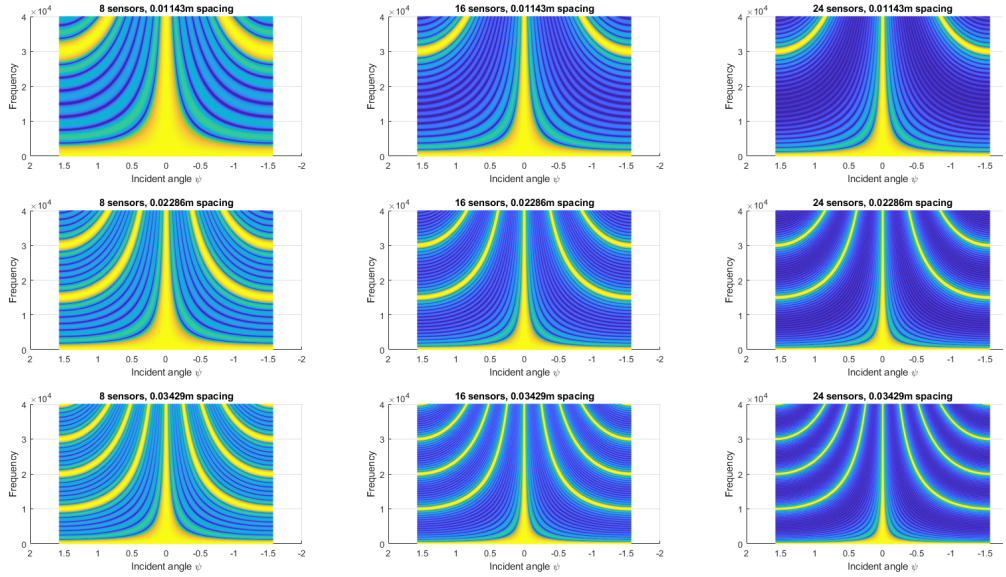

Figure 7.1: Spectral aliasing with varying number of microphones (horizontal axis) and increased spacing between adjacent sensors (vertical axis). Each plot shows frequency response vs. incident angle for 0 to 40 kHz.

The multichannel mode in the filters of the PDM to PCM conversion block makes use of a time multiplexer to convert the data from multiple microphone as shown in figure 4.3. Here. the input from each microphone is sent at every rising edge of the clock and the output for that particular channel is obtained after a number of clock cycles that is equal to the decimation rate of the filters.

CHAPTER 8

Conclusion & Future Work

CHAPTER 9

References & Acknowledgement

Bibliography

[1] 2020. URL: http://www.labbookpages.co.uk/audio/beamforming/delaySum.html.

[2] 2020. URL: http://www.labbookpages.co.uk/audio/beamforming/composite.html.

[3] D. Antonsson and M. Li. *Ultrasonic Source Localization with a Beamformer Implemented on an FPGA Using a High-density Microphone Array*. 2018.

[4] J. Benesty, J. Chen, and Y. Huang. "Conventional Beamforming Techniques". In: *Microphone Array Signal Processing*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 39–65. ISBN: 978-3-540-78612-2. DOI: 10.1007/978-3-540-78612-2_3. URL: https://doi.org/10.1007/978-3-540-78612-2_3.

[5] *CIC Compiler v4.0 LogiCORE IP Product Guide Vivado Design Suite*. 2016. URL: https://www.xilinx.com/support/documentation/ip_documentation/cic_compiler/v4_0/pg140-cic-compiler.pdf.

[6] *Field programmable gate array*. 2020. URL: https://www.xilinx.com/products/silicon-devices/fpga/what-is-an-fpga.html.

[7] *FPGA fundamentals*. 2019. URL: https://www.ni.com/sv-se/innovations/white-papers/08/fpga-fundamentals.html.

[8] S. HAUCK. "The Roles of FPGA's in Reprogrammable Systems". In: IEEE, 2019. DOI: 10.1109/5.663540.

[9] *Product data sheet SPH0641LU4H-1*. 2015. URL: https://media.digikey.com/pdf/Data%5C%20Sheets/Knowles%5C%20Acoustics%5C%20PDFs/SPH0641LU4H-1.PDF.

[10] *Top 10 FPGA advantages*. 2019. URL: https://hardwarebee.com/top-10-fpga-advantages/.