FPGA Prototype Implementation of Digital Hearing Aid from Software to Complete Hardware Design

Abdul Rehman Buzdar, Azhar Latif, Liguo Sun, Abdullah Buzdar
Department of Electronic Engineering and Information Science
University of Science and Technology of China (USTC)
Hefei, Peoples Republic of China

***Abstract*—The design and implementation of digital hearing aids requires a detailed knowledge of various digital signal processing techniques used in hearing aids like Wavelet Transforms, uniform and non-uniform Filter Banks and Fast Fourier Transform (FFT). In this paper the design and development of digital part of hearing aid is divided into three different phases. In the first phase review and Matlab simulation of various signal processing techniques used in the digital hearing aids is presented. In the second phase a software implementation was carried out and the firmware was designed for the Xilinx Microblaze softcore processor system. In the third phase everything was moved into hardware using VHDL hardware description language. The implementation was done on Xilinx Field Programmable Gate Array (FPGA) Development Board.**

***Keywords*—Hearing Aid; FPGA; CODEC; MicroBlaze; Wavelets; Filter Banks; FFT**

I. Introduction

It is in this context interesting to know how the human ear works because the underlying working principle of a hearing aid device is the same [3]. The human ear is a very complex organ. The basic structure is divided into three parts. Outer ear, middle ear and inner ear [7]. The outer ear receives the signal and directs it towards the middle ear. In the hearing aid this is usually done by directional microphones. The middle ear performs three basic functions. Firstly it acts as an impedance matching network, secondly it acts as an amplifier, thirdly and most importantly it splits the signal into different frequencies. A hearing aid device use some kind of Independence matching network, it also contains pre-amplifier stages and post-amplifiers, and also uses some kind of signal processing techniques, that is different frequency bands could be manipulated according to the patients need. The third part which is the inner ear works like a spectrum analyzer. It encodes the signal at different frequencies, makes different nerve cells resonate and transmit short pulses to the brain. In hearing aids after analyzing different frequencies we process them and then feed them to a digital analog converter in case of digital hearing aids. Fig. 1. and 2. shows a pictorial comparison between the human ear and a digital hearing aid. The frequency range of hearing for a normal human is 20 Hz to 20 kHz. Human hearing is most sensitive in the range of 1 kHz to 4 kHz [12]. There are two important parameters which effects hearing. One is loudness that is the intensity of sound and the other one is pitch, which is the frequency of the fundamental component in the sound. Human ears are designed to perceive sound on a logarithmic frequency and amplitude scale. The doubling of frequency will be perceived the same no matter what the frequency will be, for instance the doubling from 200Hz to 400Hz will be perceived the same as from 2kHz to 4kHz. Same is true for amplitude, if we double the amplitude with the same ratio no matter what the amplitude is, it will be perceived as same. We normally express sound level in dB. Normal speech is around 60dB. If the ear is exposed to 85dB the ear is at risk and the sound pressure level is harmful. Above 140dB the hearing will most certainly get damaged [13].

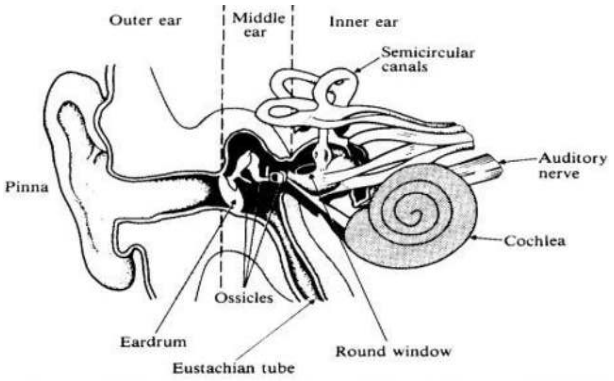

Figure 1: Structure of Human Ear

Normally the patient can suffer from three types of hearing losses [17], [18]. Patients suffering from *sensory neural loss* have some kind of damage to the auditory nerve. Some of the auditory nerves do not properly resonate with their designated frequencies and are not able to send short pulses to the brain. The other kind of loss is *conductive loss* in which the middle ear gets damaged and one does not get sound signals properly. The third kind of hearing loss is due to both conductive and sensory neural loss and is called *mixed hearing loss*. Sensory neural loss could be treated by using hearing aids, which magnifies the sound signals or cochlear implants [2], which stimulate the nerve cells directly. Nerve cells convert them to pulses which are then decoded by the brain. In case of conductive loss if the middle ear is not totally damaged hearing aid device would help by amplification of the signal in band of interest. The Hearing Aids cannot solve all hearing related issues of patients.

II. Types of Hearing Aids

We can divide hearing aids on the basis of functionality and placement [20], [22]. In *Analog Hearing Aids* every thing is composed of analog components. Analog processing is carried

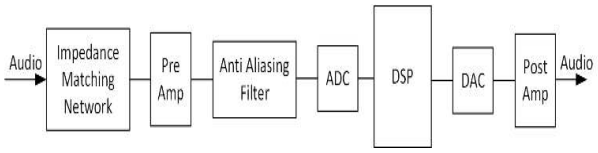

Figure 2: Block Diagram of Hearing Aid

out. The designer designs according to the patients needs adjusting the gains as needed. In *Programmable Analog Hearing Aids* the audio circuitry is built using analog circuitry but there is additional programmable control circuitry which one could use to program gain and adjust frequency settings. In *Digital Hearing Aids* almost every thing is digital from audio section to control circuitry. Very complex signal processing algorithms are there to remedy hearing losses, and also to overcome noisy environment. Humans have two ears 180 degree apart so humans can listen to sound from every direction. The sound signal could be reflected back from side walls, windows and many other objects. The human ear can detect the signal as an echo if it is received back after more than 12ms. Loose fitting of the earplug in the hearing aid could also cause feedback which is undesirable. Different echo cancellation techniques could be implemented to cancel the echo or feedback [10], [11]. Hearing aids also use compression [4], [5], [6] which basically amplify the low level sounds more than the high level sounds. If the compression is not used the hearing aid will also amplify louder sound which will be painful.

Hearing aids could also be categorized on the basis of how they are placed on the human body [21]. Following is a brief description of different hearing aids. *Completely in the Canal (CIC):* As the name suggests they are completely in the ear canal. They are not visible from outside and are for people who don't want other people to see that they are suffering from some kind of hearing loss. They are very costly. *In the Canal (ITC):* It is hardly visible from outside and is also a costly choice. It is good for conductive hearing losses. *Behind The Ear (BTE):* These kinds of hearing aids rest behind the ear. They are not costly and they are visible to other people. *Body Worn Hearing Aids:* They are most simple and bigger in size. They are the least expensive. These hearing aids consists of a case, cord and an ear mold. *Bone Anchored Hearing Aid:* Another type of hearing aid is the bone anchored hearing aid. It is implanted into the skull through operation. This kind of hearing aid can treat people having conductive loss. This kind of hearing aid transmits sound directly to the hearing nerve.

III. Signal Processing Techniques

Consider telecommunication, avionics, medical and many others, all contain very sophisticated electronics [19]. It contains everything from software running on some embedded processor to ASIC. Power, timing and area are major concerns for many of these electronic designs. Most of the electronic devices that interact with the environment or the physical world contain some kind of signal processing technique in them. It could be an algorithm or a digital filter. Digital hearing aids do contain very sophisticated algorithms and filters [20], [23]. They may also contain different transforms for time to frequency domain conversion. In all hearing aid devices one of the basic purposes is to amplify sound in specific frequency bands of interest. There are some well know digital techniques to do this. Most common are uniform filter banks, non uniform filter banks, wavelet transform and fast Fourier transform. In all of the techniques the frequencies are split into different bands. Some patients have hearing problems in the lower frequency bands whereas some patients have problems in high frequency bands. After splitting the sound into different frequency bands amplification can be done according to the patients needs. The next sections contains an overview of these signal processing techniques.

A. Uniform Filter Banks

A filter bank is nothing but a group of parallel low pass, band pass and high pass filters [30]. They are used commonly in audio systems among them in hearing aids. The basic idea is to give a common input and analyze the spectrum by splitting it into different frequency bands. In uniform filter banks all the filters have equal bandwidth. All filters use a common input signal sampled at the same frequency. Fig. 3. shows the basic architecture of the uniform filter bank. The input is common to all filter banks and the frequency split signals are then multiplied by the amplification factors and at the end the signals are added together. The best thing about uniform filter

Figure 3: Uniform Filter Bank

banks is that the architecture is very simple. It is also very easy to implement in hardware. Lets say that the sampling frequency is 8kHz and we have 4 filter banks each with a bandwidth of 1Khz. Fig. 4. shows the basic idea of band division. The bands in uniform filter banks are uniformly distributed therefore it does not provide sufficient resolution at low frequencies for the logarithmic behavior of hearing. Increasing the number of bands will fix the resolution problem at low frequencies but will give unnecessary high resolution at high frequencies. Another point to consider is the number of filter coefficients in each filter. If the number of coefficients is greater in number, no doubt the frequency response will be better but at the same time it will increase the latency which is the delay from input to output and the acceptable latency of human ear is 12ms. The dynamic switching power will also be worst in the case of uniform filter banks because of the number of multiplications and additions needed. There could also be leakage at the cut off frequencies of the filter due to the overlapping of bands.

B. Non-Uniform Filter Bank

The best thing about non-uniform filter banks is that they can exhibit the same behavior as the human ear does i.e. it can be designed logarithmically [31]. It is simpler than FFT or wavelets to implement in hardware. Fig. 5. shows the basic architecture of non uniform filter bank which is the same as for the uniform filter banks only the filter cutoff frequency changes. Lets consider a sampling frequency of 8

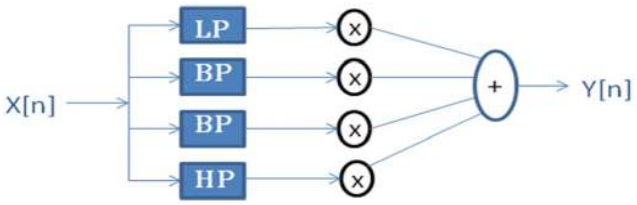

Figure 4: Filter Bank Architecture

kHz and the division of bands is non-uniform and in this case is logarithmic. The behavior is more close to the human ear as it is logarithmically divided. By using this approach better resolution could be achieved at lower frequencies. Fig. 5. shows the graphical representation of the basic idea of non-uniform filter banks. As the figure shows the resolution is better at low frequency.

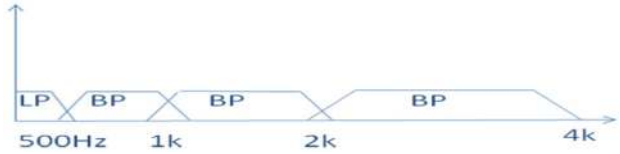

Figure 5: Non-Uniform Filter Bank

C. Wavelet Transform

Wavelet is another type of very well known transform used in hearing aids. It suits many real time applications. Wavelet works logarithmically therefore it has better frequency resolution at low frequencies [37], [38]. At the same time it has better timing resolution at higher frequencies. The architecture of the wavelets is a bit harder to implement as compared to filter banks. Fig. 6. shows the basic architecture of the wavelet transform. The signal passes through a high pass filter and a low pass filter in parallel and then gets down-sampled by a factor of two. The high pass and low pass filters split the frequency

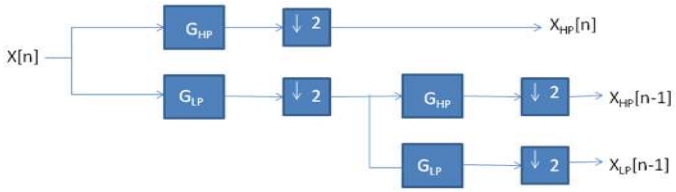

Figure 6: Wavelet Sub band Division

range into two equally sized bands. The signal from the low pass filter is again passed through a high pass filter and a low pass filter and down sampled. This is how the system gets better frequency resolution at lower frequencies. The process can go on further with more bands. For the inverse wavelet the signal is up-sampled and passes through high pass and low pass filters and the signals are then added as shown in Fig. 7.

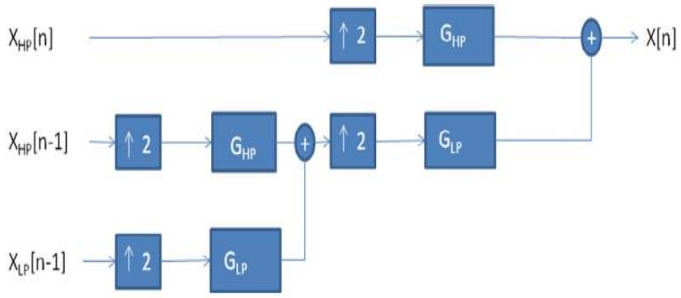

Figure 7: Wavelet Synthesis

D. Discrete Fourier Transform

The Discrete Fourier Transform (DFT) is another well known transform. It converts the time domain signal into a frequency domain signal [24], [27]. Equation (1) shows the mathematical representation of the DFT.

$$X[K] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N} \quad (1)$$

The complexity of the DFT is N^2 where N is the number of samples. For example if we take a 16 point DFT then it will require 256 complex multiplications and 256 complex additions which is really expensive in terms of computations. It would take a lot of power to compute bigger DFT's and it is not realistic to use DFT. Discrete Cosine Transform (DCT) is another choice which is basically a DFT but only contain the real part of the signal. Still DCT requires lots of computations. In terms of hardware DFT and DCT are both very expensive.

E. Fast Fourier Transform

There is a highly efficient way of calculating the DFT, the well known Fast Fourier Transform (FFT) [28], [29]. It reduces the complexity of the calculations to $N * Log_2 N$. To compute a 16 point FFT requires only 64 additions and multiplications which is much less compared to computing DFT. Increasing the number of FFT points will give even better calculation gain. The signal on which the FFT is to be applied should have $N = 2^n$ samples. FFT is really efficient in terms of less hardware i.e. it will reduce both power and area. In hearing aids the basic idea is to transform the time domain signal into the frequency domain, then amplify the frequency bands of interest and then transform the signal back to the time domain. This is exactly what a healthy cochlea [25], [26] does, a biological Fourier transform which separate sound into high and low frequency bands. To take care of some of the hearing problems FFT could be used to amplify specific band of interest. As FFT is used to transform the time domain signal into the frequency domain one could use some noise reduction algorithm [39], [40] after the FFT to manipulate frequencies for better hearing in case of noisy places like markets etc.

Directionality is another issue, after the FFT some directional algorithms could be of help to improve directionality as well [15], [16]. This could be achieved in such hearing aids where dual microphones are present [8], [9]. In many cases where voice could be reflected from walls and windows and

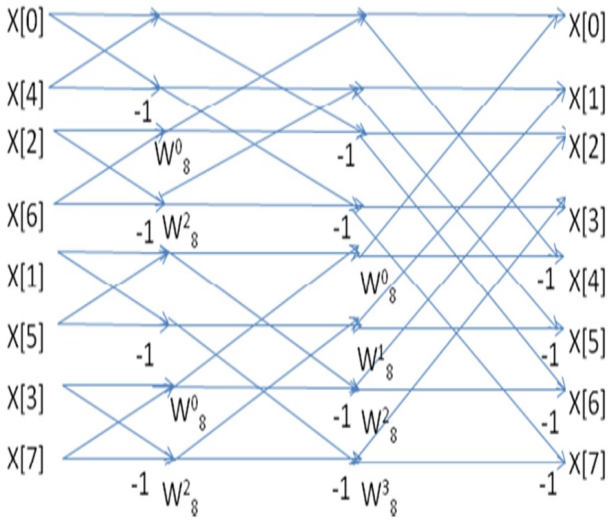

Figure 8: Decimation in Time

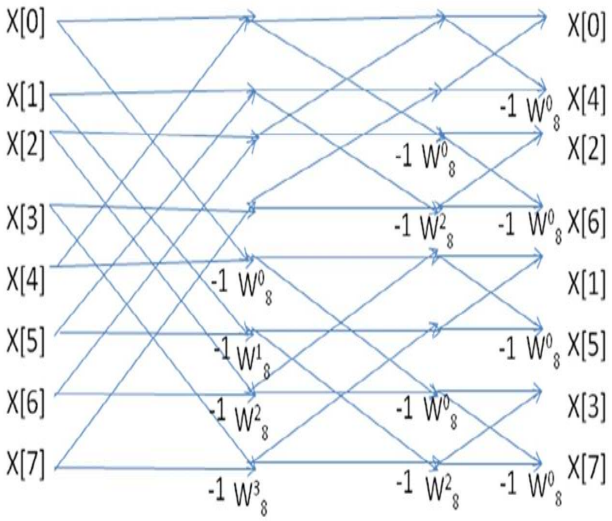

Figure 9: Decimation in Frequency

other objects may cause echo in hearing aid devices and can be noticeable, echo cancellation algorithms could be used to overcome such problems [10], [11]. The problem with FFT in hearing aids is that frequency resolution is linear so at lower frequencies the resolution will be low for a FFT with few points. It is a bit harder to implement than filter banks. FFT with non-uniform filter banks at low frequencies could give really good results. This could be achieved by diving the signal into sub bands before calculating the FFT and using more points in the lower frequency FFT which will increse the frequency resolution. Radix 2 is a very popular algorithm for computing FFT. Fig. 8. shows the FFT Radix 2 decimation in time algorithm while Fig. 9. shows FFT radix 2 decimation in frequency for N=8. An important thing to note here is the order of the input and the output samples. Decimation in time takes bit reverse inputs and generate natural order outputs whereas decimation in frequency take natural order inputs and generates an output which is in bit reversed order. After reordering analysis is performed on the resultant outputs. The FFT algorithm in case of decimation in time consists of three basic steps:

- Invert the bit order of the index which separate even and odd indexes as shown in table I.

Table I: Bit Order DFT vs FFT

Sample #	Natural Order	Reversed	Sample #
0	000	000	0
1	001	100	4
2	010	010	2
3	011	110	6
4	100	001	1
5	101	101	5
6	110	011	3
7	111	111	7

- Calculate the twiddling factor using Equation (2) below:

$$W = \sin\theta + \cos\theta \tag{2}$$

$$\theta = -2 * \pi * (\#ofGroups) * (\#ofButterflies)/N$$

- Apply a butterfly operation which will finally represent the frequencies on specific indexes as shown in Fig. 10. The butterfly operation can be mathematically

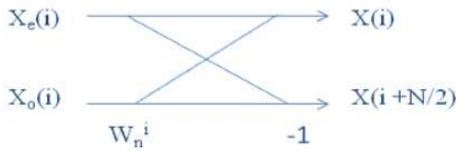

Figure 10: Butterfly Operation

represented in the form of Equations (3) and (4) below:

$$X(i) = X_e(i) + W_i^N X_o(i) \tag{3}$$

$$X(i + N/2) = X_e(i) - W_i^N X_o(i) \tag{4}$$

Real world signals are continuous in time. The problem with FFT is that it works on a finite length of the signal. Latency is an important issue in real time systems. In case of audio if the latency is more than 12ms then its not recognizable by human ears. In such cases its important to take latency into account before designing such systems. If large point FFT needs to be computed then time to gather samples and processing time should be less than 12ms. Another problem is that we have limited amount of memory and processing power. The output signal is not smooth if the sections are taken one by one. In order to make it smooth there are two methods which can be used to handle these signals. One is overlap and add and the other one is called the overlap and save method.

1) Overlap and Add Method: The basic algorithm of overlap and add consists of following four steps:

- Break down the signal of infinite length into finite blocks of length L.
- Pad zeros to these non overlapping blocks which makes the size of the new blocks L+M where L+M must be 2^n.

- Take FFT of the block including padded zeros.
- The signal can be resynthesized by IFFT and overlapping and adding the result, as shown in Fig. 11.

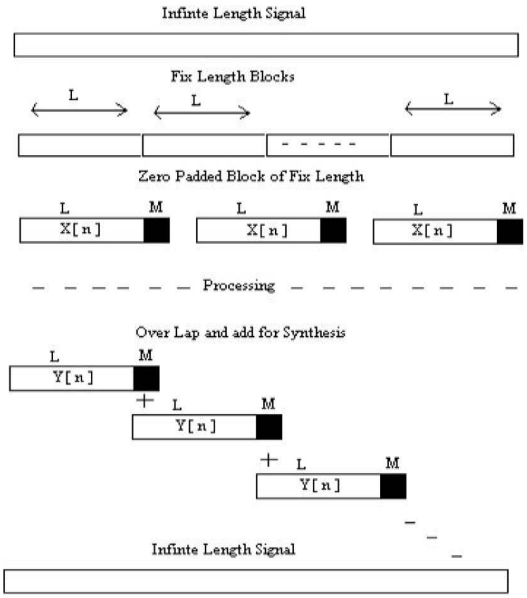

Figure 11: Overlap and Add Method

2) Overlap and Save Method: The basic algorithm of overlap and save consists of following five steps:

- Break down the signal of infinite length into finite blocks of length L.
- Place M zero samples before the first L input samples.
- In the following calculations take the last M samples from the proceeding block and place them before the L new samples.
- Do FFT analysis on L+M sample.
- The signal can be re-synthesized by IFFT and then discarding the first M values from each group of processed samples, as shown in Fig. 12.

IV. MATLAB SIMULATIONS

Basic simulations were carried out to make sure that the FFT and the IFFT results match the original signal. The IFFT signal was compared to an IDCT output as well. The IFFT was similar to the IDCT output. As the complexity of implementation of the DCT is many folds, it was decided to implement FFT as time to frequency domain transformation. Later the bands of interest could be processed according to user needs. Fig. 13. shows simulation results of the comparison between DCT and FFT. One can clearly see that the reconstructed signal is equal to the original signal.

In order to manipulate the signal in the frequency domain it is necessary to be able to analyze the frequency spectrum. Different simulations were carried out in order to understand the frequency spectrum. Fig. 14. shows the spectrum of the simulation where the sum of sin waves at 500Hz, 1200 Hz and

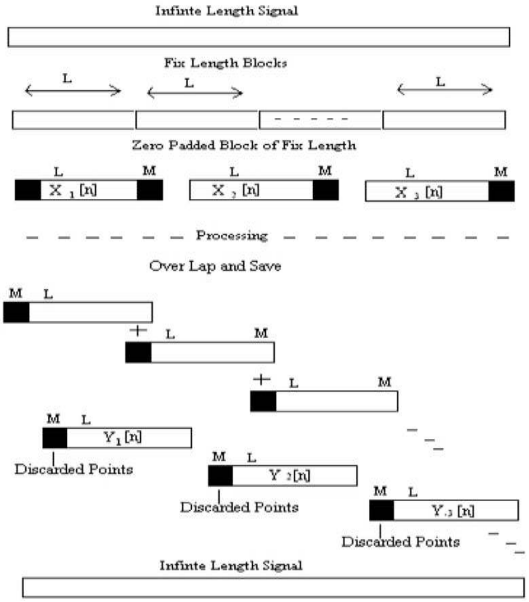

Figure 12: Overlap and Save Method

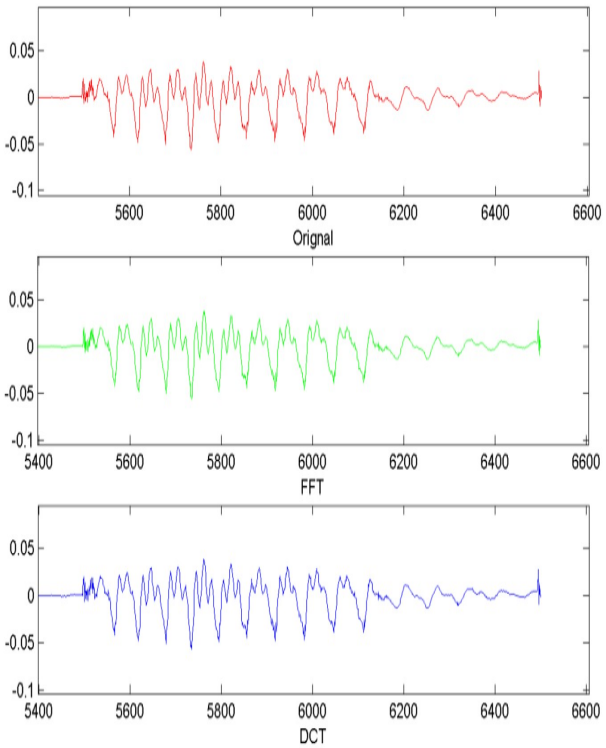

Figure 13: DCT vs FFT vs Orignal Signal

2900 Hz were summed and input to the FFT where the number of samples were 8192. The signal is sampled at 8.192Khz. As its clear from the result that the resolution is 1Hz and one can see the signal in the spectrum at exactly 500, 1200 and 2900 Hz.

Another simulation result is shown in Fig. 15. The input signal is a sine wave at 1200Hz and a 16 point FFT is carried

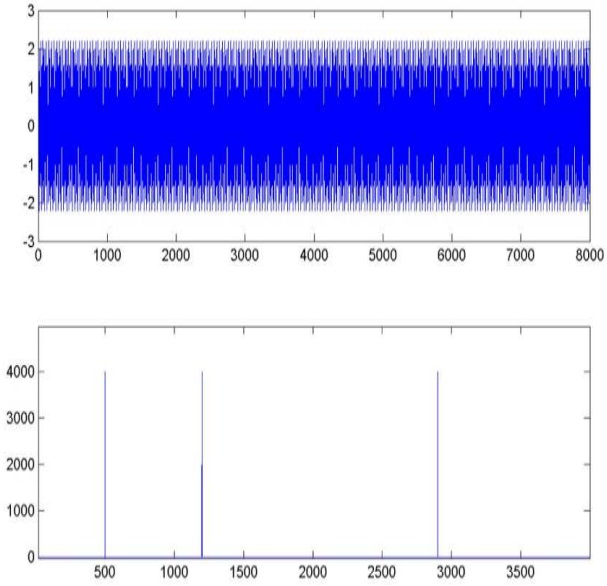

Figure 14: Spectrum of 500,1200, 2900 Hz

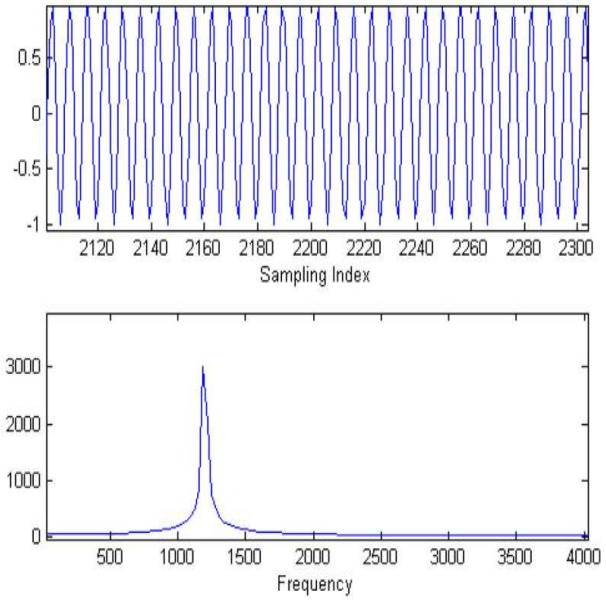

Figure 16: Spectrum of 1200 Hz with 256 points

out. The signal is sampled at 8 kHz. The resolution is 500Hz. It is obvious from the Fig. 15 that the frequency resolution is not very good. The spectrum doesn't show the signal located at 1 kHz because there is no frequency slot there. The closest frequency slots are at 1 kHz and 1.5 kHz.

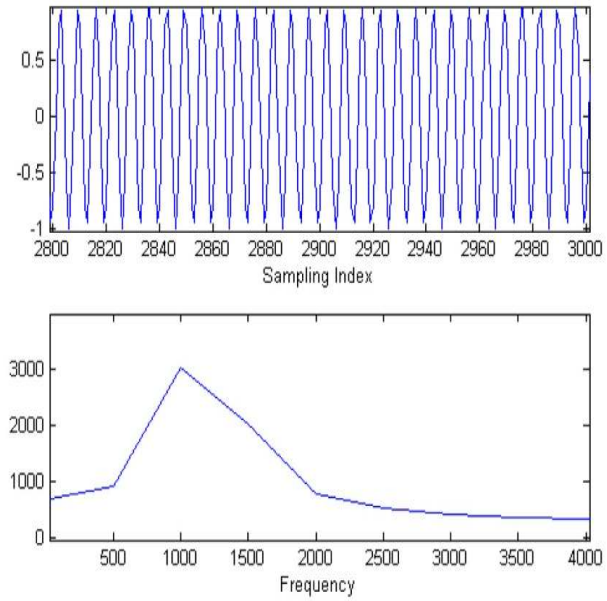

Figure 15: Spectrum of 1200 Hz with 16 points

Spectrum analysis of Fig. 16. shows that the resolution in frequency improves if we take more points in the FFT. The same signal of 1200Hz sine wave was fed to the FFT at a sampling frequency of 8 kHz using a 256 point FFT. Fig. 16. clearly shows that the resolution has increased to 31.25 Hz. It is obvious that the signal is located near 1200 Hz on the frequency scale.

V. Basic Description of System

All of the digital hearing aids must contain some common basic blocks [1], [3]. It includes microphone, impedance matching network, preamplifiers, antialiasing filters, analog to digital converter, some kind of signal processing technique using custom hardware or an embedded processor, digital to analog converter, post amplifier, speaker. Fig. 2. shows a basic diagram of a digital hearing aid. Normally directional microphones are used in hearing aids. Some kind of impendence matching network is designed to deliver proper signal level from microphone to the preamplifier stage. The preamplifier stage amplifies the input signal and feds the signal to the analog to digital converter. Every digital hearing aid also have an antialiasing filter before the ADC. The digitized signal is then processed. Modern digital hearing aids implement very complex algorithms for feedback and echo cancellation[10], [11]. Directivity is improved using different techniques[14], [15], [16]. Adaptive filtering techniques are used so that the hearing aid device should work in an intelligent manner i.e. it should be able to work in noisy places like markets and crowded places. The basic purpose of any hearing aid device is to amplify the signal in the bands of interest. All the latest digital devices come with some kind of user interface so that the user can adjust the gains according to the patient's needs, then the processing signal is fed to a digital to analog converter or pulse width modulator(PWM). PWM is the simplest type of DAC. The time domain discrete signal is fed to the PWM, the duty cycle of the output pulses change and then pass through a low pass analog filter. The simple architecture of PWM reduces the complexity of hardware. A speaker is used at the end of the device chain.

For the purpose of implementation the Digilent XUP II Virtex-II Pro system [32] is used. The FPGA on the board is a Xilinx XC2VP30 [33] with 30,816 logic cells. It contains 136 18bit multipliers and 2448 kb of block RAM. The board contains an audio codec the LM4550 chip from National

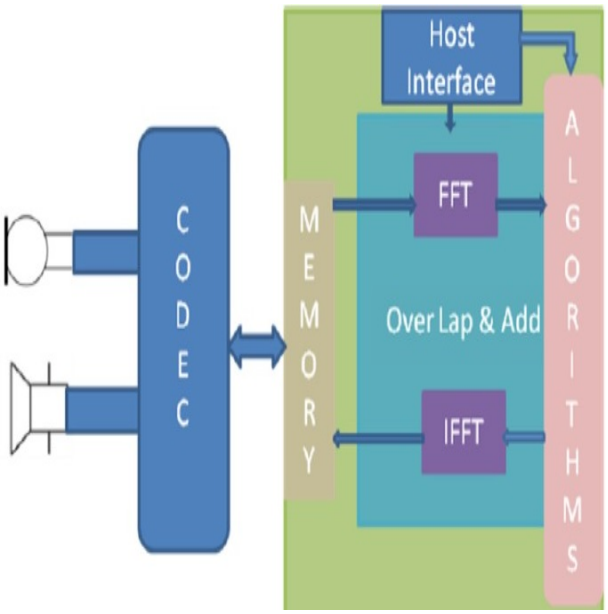

Figure 17: System Level Diagram of Hearing Aid

Semiconductor [34], which is used as an ADC and DAC. The board contains an amplifier at the output stage as well. The board is running at 100MHz. As the FFT is easier to implement than the DCT and needs very few computations FFT became the choice of implementation. For the purpose of prototyping the signal is fed through a microphone to the input of channel one of the codec. The digitized output from the ADC of the codec is fed to memory which is basically a FIFO. Fig. 17. shows the system level view of Digital Hearing Aid implementation. After receiving samples they are divided into blocks of 16 samples. The 16 sample block is then fed to a FFT calculation block which transforms the information from time domain to frequency domain. After this the bands of interest are amplified. The gain values in this prototype were changed using the push buttons of FPGA board. Two kinds of host interfaces were also tried. First a USB phy host interface was tried. A free open core source implementation was tried but it did not match the exact electrical characteristics of the USB phy interface. Another serial UART implementation was later tried at 9600 bps. After converting the audio signal from time domain to frequency domain, different kinds of complex signal processing algorithms can be used to improve hearing. It includes echo and feedback cancellation, noise reduction, filtering, adaptive noise cancellation etc [39], [40]. The implementation of the algorithm will of course affect the implementation complexity, latency, power and area. To synthesize the signal back to the time domain an inverse fast Fourier transform (IFFT) is used. The 16 samples were fed to the DAC of the codec. The board contains an amplifier at the output stage. The output is then fed to a speaker. Fig. 17. shows the block level diagram of the system implemented.

VI. Software Implementation

On the basis of the simulation results the whole system was implemented in software using the Xilinx Microblaze softcore processor system [35]. It could be of great help in some cases to implement part of the design in software and part of it in hardware to achieve acceleration [41], [42], [43], [44], [45]. The program code was written in C programming. The implementation was further divided into two phases. In the first phase the MicroBlaze processor general purpose input output (GPIO) was used to configure the LM4550 codec [34]. Due to the sequential nature of the MicroBlaze processor, it was not possible to configure the LM4550 codec properly. In the second phase a separate wrapper for the LM4550 codec was added as a peripheral which took care of the timing issues. The Peripheral Local Bus (PLB) was used to connect the LM4550 codec wrapper to the MicroBlaze processor. A FIFO was used to synchronize the data transfer between the MicroBlaze processor and the LM4550 codec wrapper. The FFT algorithm was implemented in the MicroBlaze processor. Both blocking and non-blocking approaches were tried. Blocking means that

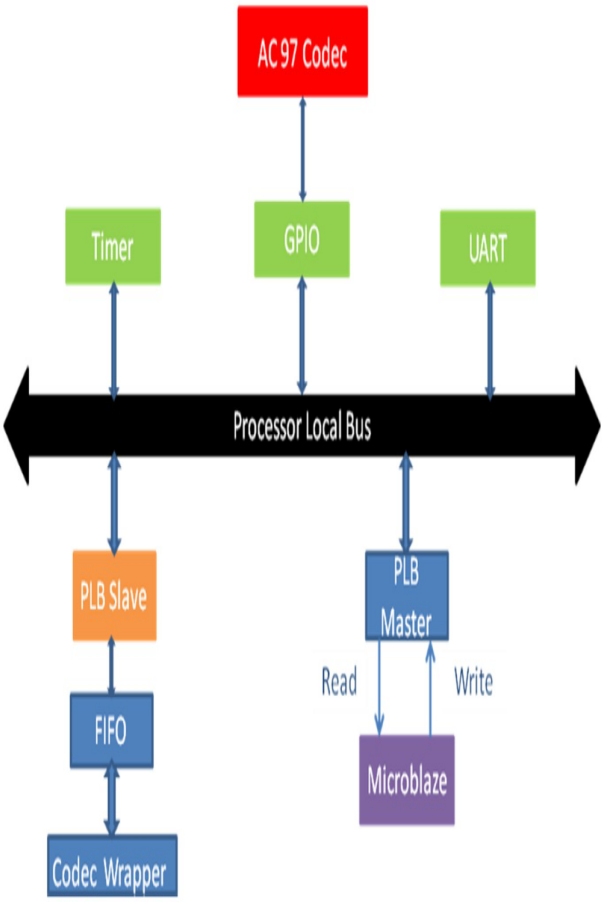

Figure 18: Block Diagram of Software Implementation

data is recieved through the data bus and is stored in an array before it is processed further. Non blocking means that data is recieved and processed continously. The LM4550 codec wrapper as a user peripheral was not only responsible for programming of the codec but also for reading digitized inputs from the codec chip and writing output samples to the codec. Fig. 18. shows the basic architecture of the software implementation. Various problems were faced during implementation. It required lots of effort to debug various issues using the provided debugger. The slow behavior of the debugger was one

of the major reasons. The codec's serial clock was at 12.288 Mhz whereas the debugger response was very slow. It became nearly imposible to debug the serial protocol in real time, that was one of the main reasons to add the LM4550 codec wrapper as a peripheral. As printf functions takes to much memory, almost 1k, Xilinx has provided a special function to print to the terminal but it was still taking too much memory and was printing the wrong results. After some debugging it was found that if one increases the stack size it could solve the problem. The FFT algorithm was working fine, but the stack size was the problem. Another really big issue was to connect the GPIO's to the user hardware. The solution to the problem was really tedious. It took some manual changes to some of the design files including user logic in the hdl folder and the file with extension pao in the data folder. GPIOs and user peripherals are connected to the PLB, therefore if a user peripheral wants to communicate with some external hardware, in this case the LM4550 codec, then the user peripheral has to go through the PLB to communicate with the GPIO's. The flow of communication goes like this. User peripheral writes command and data to the PLB FIFO which writes to the PLB bus. The GPIO's are also connected to the PLB so data is read and written to the external LM4550 codec chip. The data read from the LM4550 codec is sent to the FFT for frequency domain conversion and after processing the bands of interest, an IFFT is carried out and written back to the codec through the PLB and GPIOs.

VII. Hardware Implementation

The main building blocks in the hardware implementation includes the Xilinx FFT core [36] and the Xilinx Dual port RAM [33]. Fig. 19. shows the block diagram of hardware implementation of Digital Hearing Aid. The architecture chosen for the Xilinx FFT core was pipelined streaming I/O 16 point FFT. Input resolution of the codec is 20 bit. The output resolution is 25 bit but the DAC only uses 20 bits. The order of the output was natural. The same core was used for both the FFT and IFFT. Latency of the system is 333usec which is far less then the limit of 10ms for hearing aids, the latency recognizable by normal human ear. There is still lots of room and time to implement other complex algorithms in hardware for instance to reduce noise, echo cancellation [39], [40], improving directionality [15] etc. Two Xilinx Dual port RAMs were used. The width of the RAMs was 20 bits and the Depth of the RAMs was 32 samples. Both RAMs were configured in Read After Write (RAW) mode.To configure the LM4550 codec chip a wrapper was used which not only configure the chip but also receives the samples from the ADC of the codec and output serial data to the DAC of the codec over a five wire serial interface. It includes serial data in, serial data out, reset for the LM4550 codec chip reset and serial clock at 12.288 MHz. The LM4550 codec chip could be programmed to sample at a rate from 4 kHz to 48 kHz in steps of 1 kHz. Different sampling rates in the range of 16kHz to 48kHz were tried during implementation. The LM4550 codec contains a sigma delta ADC whose dynamic range is 90dB. The dynamic range of the DAC is 89dB. The LM4550 codec gain is programmable and could be programmed from 0dB to 22.5dB in 1.5dB steps. The board is running at 100MHz. The LM4550 codec chip provides a 5 line serial programmable interface. At the very beginning reset is applied to the LM4550

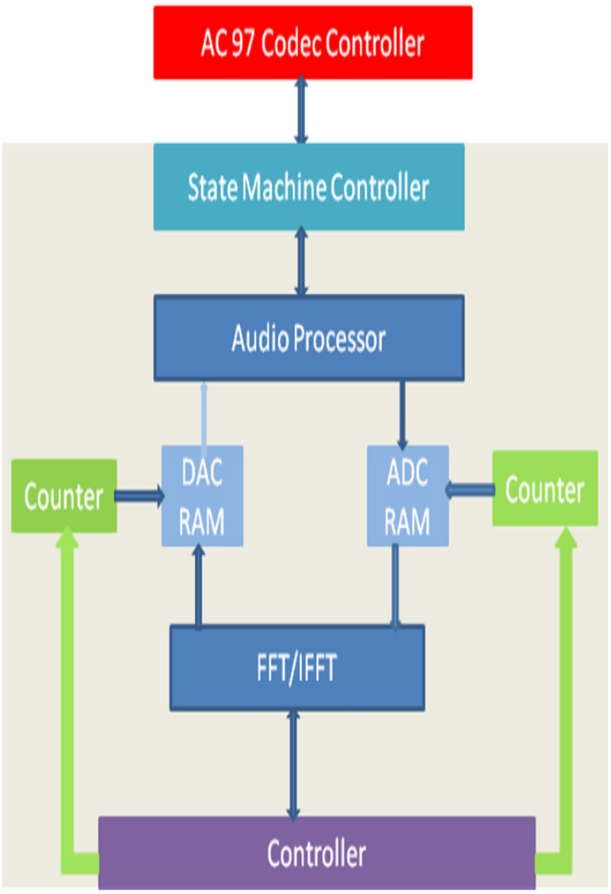

Figure 19: Block Diagram of Hardware Implementation

codec which resets all registers and internal circuitry to a default state. At the start of each frame a serial synchronization pulse goes from low to high which is used as a soft reset for the LM4550 codec chip. It is used to clear the value in the status register of the LM4550 codec. Each bit is received and sent on the bit clock over the serial lines of the interface. The frequency of the bit clock is 12.288MHz. Each serial programmable protocol frame consists of 13 slots. Each frame consists of 256 bits. Slot 0 consists of 16 bits while the rest of the slots consists of 20 bits each. Each bit in the slots has a specific purpose. They indicate status of the frame or data. For detailed description refer to the data sheet of LM4550 [34]. Fig. 19. shows a block level diagram of the main modules of hardware implementation of Hearing Aid.

After each hard reset and configuration of the LM4550 controller the audio processor block gathers samples from the LM4550 ADC. The samples are then fed to Xilinx Dual port RAMs. A counter was implemented which keeps track of if 16 samples have arrived. As soon as 16 samples are gathered these samples are fed to the FFT block for time domain to frequency domain conversion. After that the FFT bands of interest can be amplified. As the FFT core is running at 100 MHz and a 16 point FFT is calculated, there is plenty of time available to implement complex signal processing algorithm for better hearing. A controller was implemented which keeps track of the control signals from the core. As soon as the transform

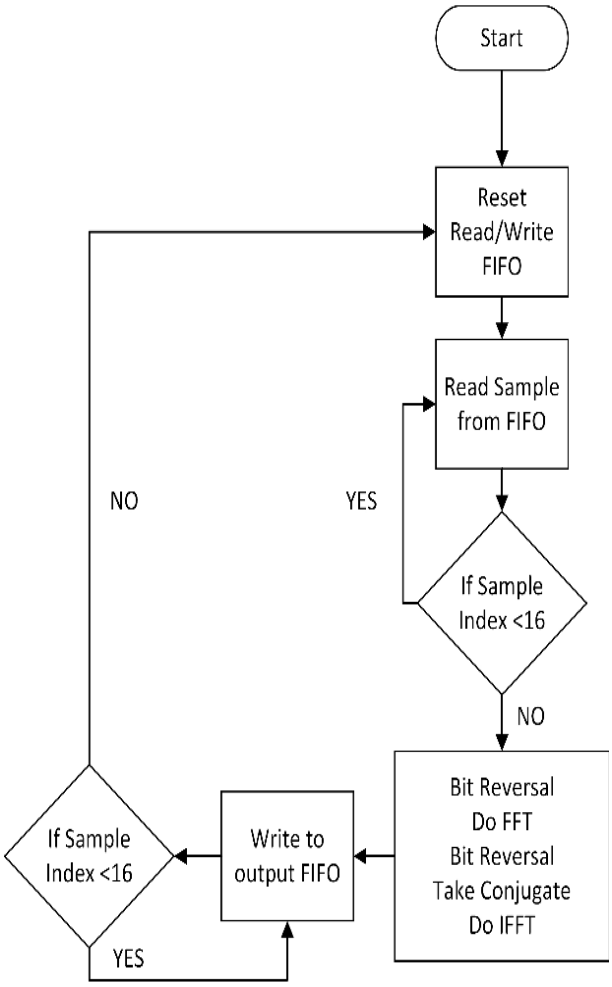

Figure 20: Flow chart of Hardware Implementation

is completed the core asserts a done signal. The same core is then used to calculate the IFFT. The core has an input signal which tells the core to calculate forward transform or reverse transform. The core needs three cycles delay before it is fed back to the FFT core to calculate the inverse transform. Three flip flops were added to take care of this issue. As soon as the inverse Fourier transform is finished the core asserts the done signal again. As the natural order for the output was selected therefore there was no need for reordering of the output samples. The samples are then fed to the RAM block. The 16 processed audio samples are then serially fed back to the LM4550 audio codec. Basic Simulation of the FFT was carried out in Modelsim. Fig. 20 shows the flow chart of hardware implementation. The results of the core were verified by comparing the results in Matlab. Debugging was carried out using both ChipScope and Modelsim. Average resource utilization of the FPGA is about 12% which includes Flip-Flops, Shift Registers, BRAMS, and Multipliers.

VIII. Conclusion

The main purpose of the paper was to implement a prototype for a digital hearing aid. In the first phase different signal processing techniques were analyzed which includes wavelet transform, uniform filter banks, non-uniform filter banks, DCT and FFT. In the second stage a software implementation was carried out on Xilinx Microblaze soft core processor system. It could be of great help in some cases to implement part of the design in software and part of it in hardware to achieve acceleration. In the third stage everything was implemented in hardware. Power, area and timing are the main issues of modern electronics. These are more critical in the case of hearing aids. When considering ASIC some kind of techniques for power reduction should be used for instance gate clocking could be used. Dynamic voltage and frequency scaling could save a lot of power. Area is another important issue. Register size, multipliers, adders, resolution filter coefficient directly effect area. More hardware can also result in more power consumption. The system could be designed to run at the minimum possible frequency which will reduce dynamic switching power. Different signal processing algorithms like echo cancellation, feedback compensation, noise reduction, adaptive filtering, better directionality etc could be implemented to make hearing aid work better.

IX. Acknowledgments

This work is partially supported by the Chinese Academic of Sciences (CAS) and The World Academy of Sciences (TWAS).

References

[1] Desloge, J.G., Rabinowitz, W.M. and Zurek, P.M., “Microphone-array hearing aids with binaural output. I. Fixed-processing systems,” IEEE Trans. Speech and Audio Processing, vol. 5, pp. 529-542, Nov. 1997.

[2] L. Watts, D.A. Kerns, R.F. Lyon, C.A. Mead,“Improved implementation of the silicon cochlea,” IEEE Journal. Solid-State Circuits, vol. 27, no. 5, pp. 692-700, May 1992.

[3] Marion A. Hersh and Michael A. Johnson, Assistive Technology for the hearing impaired, Deaf and Deafblind, London, Springer-Verlag, 2003.

[4] Xulai Cao, Linghua Zhang, Wanzhen Gao, “A new multi-channel loudness compensation method based on high frequency compression and shift for digital hearing aids”, IEEE International Conference on Wireless Communications and Signal Processing (WCSP), Oct. 2013, pp. 1-6.

[5] Kuo-Chiang Chang, Yu-Ting Kuo, Tay-Jyi Lin, Chih-Wei Liu, “Complexity-effective dynamic range compression for digital hearing aids”, IEEE International Symposium on Circuits and Systems (ISCAS), May. 2010, pp. 2378-2381.

[6] M.A. Trenas, J.C. Rutledge, N.A. Whitmal, “Wavelet-based noise reduction and compression for hearing aids”, IEEE International Conference on Engineering in Medicine and Biology, Oct. 1999.

[7] A F. Jahn, J SantosSacchi. (2001). Physiology of the Ear. Singular; 2nd edition. Columbia.

[8] Guangji Shi, P. Aarabi, Hui Jiang, “Phase-Based Dual-Microphone Speech Enhancement Using A Prior Speech Model,” IEEE Trans. Audio, Speech, and Language Processing, vol. 15, no. 1, pp. 109-118, Jan. 2007.

[9] O. Hoshuyama, “Dual-microphone echo canceller for suppressing loud nonlinear echo”, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), March. 2012, pp. 181-184.

[10] C. Paleologu, J. Benesty, S. Ciochina, “Study of the General Kalman Filter for Echo Cancellation,” IEEE Trans. Audio, Speech, and Language Processing, vol. 21, no. 8, pp. 1539-1549, Feb. 2013.

[11] P. Loganathan, A.W.H. Khong,P.A. Naylor, “A Class of Sparseness-Controlled Algorithms for Echo Cancellation,” IEEE Trans. Audio, Speech, and Language Processing, vol. 17, no. 8, pp. 1591-1601, Nov. 2009.

[12] Steven W. Smith, The Scientist and Engineer’s Guide to Digital Signal Processing, Second edition, 1999.

[13] Health and Safety Executive UK. [Online]. Available: http://www.hse.gov.uk/noise/faq.htm

[14] J. Woodruff, DeLiang Wang, "Directionality-based speech enhancement for hearing aids", IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP), May. 2011, pp. 297-300.

[15] T. Van de Bogaert, J. Wouters, T.J. Klasen, M. Moonen, "Distortion of interaural time cues by directional noise reduction systems in modern digital hearing aids", IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, Oct. 2005, pp. 57-60.

[16] M. Aubreville, S. Petrausch, "Directionality assessment of adaptive binaural beamforming with noise suppression in hearing aids", IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), April. 2015, pp. 211-215.

[17] American Hearing Research Foundation. [Online]. Available: http://www.american-hearing.org/disorders/hearing-loss/

[18] Hearing Loss Association of America. [Online]. Available: http://www.hearingloss.org/content/types-causes-and-treatment

[19] D. George Gata, Walter Sjursen, "A 1.1-V 270uA mixed-signal hearing aid chip," IEEE J. Solid-State Circuits, vol.37, pp. 1670-1678, Dec. 2002.

[20] Robert E. Sandlin, Textbook of Hearing Aid Amplification Technical and Clinical Considerations, second edition, San Diego, California, Singular Publishing, 1999.

[21] Michael Valente, Hearing Aids: Standards, Options, and Limitations, New York, Thieme Medical Publisher, 1996.

[22] Robert A. Goidenberg, Hearing Aids a manual for clinicians, New York, Lippincoot-Raven Publisher, 1996

[23] H.G.McAllister, Hearing aids-a development with digital signal processing devices, IEEE Computing and Control Engineering Journal, pp 283-229, Dec 1995.

[24] A. V. Oppenheim and R. W. Schafer, Discrete-Time Signal Processing. Prentice-Hall, 1999.

[25] Wang Shiwei, T.J. Koickal, A. Hamilton, R. Cheung, L.S Smith, "A Bio-Realistic Analog CMOS Cochlea Filter With High Tunability and Ultra-Steep Roll-Off," IEEE Trans. Biomedical Circuits and Systems, vol. 9, no. 3, pp. 297-311, July. 2014.

[26] M. Nouri, A. Ahmadi, S. Alirezaee, G. Karimi, M. Ahmadi, D. Abbott, "A Hopf Resonator for 2-D Artificial Cochlea: Piece wise Linear Model and Digital Implementation," IEEE Trans. Circuits and Systems I, vol. 62, no. 4, pp. 1117-1125, April 2015.

[27] Zhongde Wang, "Fast algorithms for the discrete W transform and for the discrete Fourier transform," IEEE Trans. Acoustics, Speech and Signal Processing, vol. 32, no. 4, pp. 803-816, Aug. 1984.

[28] J.A. Glassman, "A Generalization of the Fast Fourier Transform," IEEE Trans. Computers, vol. C-19, no. 2, pp. 105-116, Feb. 1970.

[29] Mitra, Sanjit Kumar, and Yonghong Kuo. Digital signal processing: a computer-based approach. New York: McGraw-Hill, 2006.

[30] M. Harteneck, S. Weiss, and R. W. Stewart, "Design of near perfect reconstruction oversampled filter banks for subband adaptive filters," IEEE Trans. Circuits Syst., vol. 46, no. 8, pp. 1081-1086, Aug. 1999.

[31] A.Pandey, V. John Mathews, "Low-Delay Signal Processing for Digital Hearing Aids," IEEE Trans. Speech Audio Process., vol. 19, no. 4, pp. 699-710, July. 2010.

[32] Digilent Inc. XUP II Virtex-II Pro system. [Online]. Available: http://www.digilentinc.com/.

[33] Xilinx Inc. FPGA Design Tools. Silicon Devices. [Online]. Available: http://www.xilinx.com/

[34] National Semiconductor LM4550 Data Sheet: [Online]. Available: http://www.xilinx.com/products/boards/ml510/ datasheets/lm4550.pdf

[35] Xilinx. MicroBlaze Soft Processor Core. [Online]. Available: http://www.xilinx.com/tools/microblaze.htm

[36] Xilinx Fast Fourier Transform (FFT) CORE. [Online]. Available: http://www.xilinx.com/products/intellectual-property/fft.html

[37] F.C.A.Fernandes, R.L.C.van Spaendonck, C.S.Burrus, "A new framework for complex wavelet transforms," IEEE Trans. Signal Processing, vol. 51, no. 7, pp. 1825-1837, July. 2003.

[38] A.M.Grigoryan, "Fourier transform representation by frequency-time wavelets," IEEE Trans. Signal Processing, vol. 53, no. 7, pp. 2489-2497, July. 2005.

[39] Min Li, Hector G. McAllister, Norman D. Black and Trina Adrian De Perez, "Perceptual Time-Frequency subtraction algorithm for Noise Reduction in Hearing Aids," IEEE Trans. Biomedical Engineering, vol. 48, pp. 978-988, Sept. 2001.

[40] Ram Singh Rana, Zhang Liang, Tang Bin and Garg Hari Krishna, "An enhanced method and behavioral model for noise cancellation in audio devices", IEEE International Workshop on Biomedical Circuits and Systems, Singapore, pp S2.6-11-S2.6-14, Dec. 1-3, 2004.

[41] Abdul Rehman Buzdar, Liguo Sun, Azhar Latif and Abdullah Buzdar, "Instruction Decompressor Design for a VLIW Processor", Informacije MIDEM-Journal of Microelectronics, Electronic Components and Materials Vol. 45, No. 4 (2015).

[42] Abdul Rehman Buzdar, Liguo Sun, Azhar Latif and Abdullah Buzdar, "Distance and Speed Measurements using FPGA and ASIC on a high data rate system" International Journal of Advanced Computer Science and Applications(IJACSA), 6(10), 2015, 273-282.

[43] M. Imai, "Embedded tutorial: hardware/software codesign", IEEE Asia and South Pacific Design Automation Conference (ASP-DAC), Jan. 1999.

[44] J. Noguera, R.M. Badia, "HW/SW codesign techniques for dynamically reconfigurable architectures" IEEE Trans. Very Large Scale Integration (VLSI) Systems, vol. 10, no. 4, pp. 399-415, Aug. 2002.

[45] M. D. Edwards, et al., "Acceleration of software algorithms using hardware/software co-design techniques", J. Syst. Architecture, vol. 42, no. 9/10, pp.1997.