Implementation of epoch detection and its application on FPGA

A thesis submitted in partial fulfillment
of the requirements for the degree of

*Master of Science in **Electronics and Communication Engineering** by Research*

by

Syed Abdul Jabbar
2020702022
abdul.jabbar@research.iiit.ac.in

Advisor: Dr. Anil Kumar Vuppala

International Institute of Information Technology Hyderabad
500 032, India

April 2024

International Institute of Information Technology Hyderabad
Hyderabad, India

CERTIFICATE

This is to certify that work presented in this thesis proposal titled ***Implementation of epoch detection and its application on FPGA*** by *Syed Abdul Jabbar* has been carried out under my supervision and is not submitted elsewhere for a degree.

_____ _____

Date Advisor: Dr. Anil Kumar Vuppala

To **Family,**

and

My guide

Dr. Anil Kumar Vuppala

Acknowledgments

First and foremost, I would like to express my sincere gratitude and deepest respect to my supervisor, Dr. Anil Kumar Vuppala. His suggestions, guidance and support throughout my Master's journey have been invaluable. I sincerely thank him for encouraging me and build my confidence at every step. I will always be grateful for his support.

I would like to thank Dr. Syed Azeemuddin, for his every support and encouragement. I appreciate all his valuable suggestions, motivational talks, and the assistance he provided during the challenging times throughout my journey.

I also like to thank Dr. Chiranjeevi Yarra, for his efforts to make all lab members active and valuable.

I would like to thank to all my friends, seniors, juniors who made my life memorable at IIIT-H.

I would also like to thank to Huzaifa, Javid, Salman, Kunal, and all other CVEST lab members who showed their support and help me out in my tough times at the early stage of my research.

I also want to thank to Krishna sir, Purva maam, Ganesh sir, Vamshi sir, Nayan all my lab members who support encourage and made my days memorable at IIIT-Hyderabad.

Special thanks to Purva maam and krishna sir who helped and support almost every time. I am grateful to Purva ma'am for her support, motivation, encouragement, and assistance from the beginning of my research. She never hesitated or showed frustration while supporting me. I am grateful to have these kind of lab members.

I also want to express my gratitude to all the teachers who supported me morally and technically.

Finally, thanks to my family from bottom of my heart for their efforts, support, encouragement and selfless support throughout my journey.

Abstract

Epoch is the instant at which significant excitation of vocal tract filter takes place during the production of speech phonation. The extraction of epochs helps in speech enhancement and multi-speaker separation. For most voiced speech, the significant excitation takes place around the instant of glottal closure. Several methods have been proposed for estimating the instants of glottal closure from speech signal without using the electro-glottograph (EGG) signal. These methods are based on short-time energy of the speech signal, predictability of an all pole linear predictor, and properties of group-delay. From lot of different methods, Zero Frequency Filtering (ZFF) and Zero Phase Zero Frequency Resonator (ZP-ZFR) is one of the the best and simplest method to find prominent locations of epochs which gives highest accuracy detection rate, and it also outperforms many other parameters.

This thesis work focuses on the implementation of the ZP-ZFR algorithm and its application on Field Programmable Gate Array (FPGA). The ZP-ZFR algorithm explains its principles and advantages over the traditional ZFF algorithm. The implementation details of both algorithms on FPGA are presented, including the software simulation phase using MATLAB and the subsequent implementation using FPGA boards. Moreover, the application ZP-ZFR based Voiced Activity Detector (VAD) is implemented. The primary objective is to evaluate the effectiveness and efficiency of the ZP-ZFR algorithm and detecting voiced and unvoiced segments in speech signals which are useful for real time applications. FPGAs are well-suited for also many applications, such as object detection, tracking, and recognition, radar systems, digital signal processing (DSP) and mainly for vision based applications. FPGA implementation is carried out because, these are inherently parallel devices, allowing multiple operations to be executed simultaneously. This makes them well-suited for applications that require real-time processing, such as speech recognition and speech synthesis etc. FPGAs can also be customized to accelerate specific tasks by implementing dedicated hardware. This is highly beneficial for applications requiring real-time processing or demanding computational tasks. It is also known for low-latency operations and can be more power-efficient for specific tasks.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Speech production is the process by which thoughts are translated into speech. It is a fundamental and instinctual form of human interaction, playing a big part in our daily communication. We use speech production to communicate with others, share information, express emotions. Researchers in the field of speech technology are dedicated to advancing our ability to comprehend and replicate this unique human capability. Researchers across various fields, including linguistics, artificial intelligence, study speech production to advance our understanding of language. They are trying to work on developing systems that not only understand human speech but also engage in meaningful conversations with people.

Human speech production involves highly complex motor tasks in producing various sounds. It requires airflow from the lungs (respiration) to be phonated through the vocal folds of the larynx (phonation) and resonated in vocal cavities shaped by the jaw, soft palate, lips, tongue, and other articulators (articulation) [1]. The speech production system is divided into three parts: sub-glottal system, larynx, and supra-glottal system (structures and airways above the larynx also known as a vocal tract). During Speech production, a constriction is usually formed in the airways at the level of the vocal folds, located within the larynx. This constricted region, which is just a few millimetres long, is called the glottis, and it forms the dividing line between the sub-glottal system and the supra-glottal system. For the production of the most speech sounds, the sub-glottal system provides the energy in the air flow, and the laryngeal and supra-glottal structures are responsible for the modulation of the air flow to produce audible sound. As we shall see, the energy for some sounds in obtained by trapping air within an enclosed space above the larynx, and expanding or contracting the volume of the space [2].

In the speech signal processing, an epoch is the instant of significant excitation of the vocal-tract system during production of speech. In most voiced speech, the most prominent excitation occurs around the glottal closure instant. Extracting epochs from speech poses a challenge due to the time-varying nature of both the source and the vocal tract system. Most epoch extraction methods aim to eliminate the characteristics of the vocal tract system to amplify the excitation characteristics in the residual signal. The performance of such methods depends critically on our ability to model the system [3]. From lot of methods, Zero-phase Zero Frequency Resonator (ZP-ZFR) [4], and a base line method Zero Frequency Filter (ZFF) [3], are one of the best and simplest methods to find prominent locations of epochs which gives highest accuracy detection rate and lowest false alarm rate. The ZP-ZFR

algorithm also outperforms many other parameters. Therefore, the implementation of ZP-ZFR has been proposed on FPGA using verilog, which is a Hardware Description Language (HDL). Implementing algorithms on an FPGA allows for real-time processing of audio signals with low latency.

1.1 Problem Statement

The primary objective is to implement stable epoch detection method and its application using FPGA. Stability is a crucial consideration when implementing systems, including those on FPGA's. The stability ensures consistent and reliable performance of the system. By achieving stability, the system can consistently and accurately identify epoch locations. To achieve stability, some approaches has been proposed in the recent years like, finite impulse response (FIR) implementations of zero frequency filter [5], [6], [7]. But all these methods require higher filter order and they are complex in design. Later, the ZP-ZFR technique is proposed which is an infinite impulse response (IIR) filter which requires lower filter order [4]. It is a stable version of ZFF, which gives better performance than ZFF, due to the location of poles used in implementation of ZP-ZFR is taken inside the unit circle. Up to our knowledge ZP-ZFR based approach is not yet implemented using FPGA. Therefore, the thesis primary objective is to implement stable epoch detection method and its application ZP-ZFR based Voice Activity Detector (VAD) using HDL verilog.

1.2 Contribution

The contributions in this thesis are as follows:

- The stable epoch detection algorithm ZP-ZFR is implemented using FPGA.
- Implemented ZP-ZFR based VAD application using HDL verilog, which can use for many applications like speech coding, voice controlled systems, speech feature extraction, etc.
- In addition to ZP-ZFR, we also considered and implemented the ZFF method in our study. ZFF serves as a baseline method against which we compare the performance of ZP-ZFR.

1.3 Thesis Organisation

This thesis is organized in the following manner: Chapter 1 discuss about speech production and presents the problem that the research will address. Chapter 2 gives an overview on epoch extraction techniques and helps to understand the proposed work along with the existing literature. Chapter 3 explains the epoch extraction algorithm ZP-ZFR in detail and its implementation using FPGA, along with the baseline method. Subsequently, we present the dataset used for extraction and implementation, along with the assessment and results. Chapter 4 presents the application VAD using ZP-ZFR algorithm. Both the software and HDL implementation of VAD using ZP-ZFR are presented, along with the

baseline method, ZFF. Later we describe the software and hardware results of ZFF and ZP-ZFR based VAD, along with the dataset. Lastly, Chapter 5 concludes the thesis and suggests a future research work.

Chapter 2

Literature Review

2.1 Introduction

During the speech production, significant excitation to the vocal tract system happens around the epoch locations. These epoch locations, derived from the speech signal which are useful in many real-time applications. Porting epoch based algorithms to FPGA is beneficial in applications such as telecommunication systems and voice-controlled devices etc. Nowadays, many digital electronic systems are implemented on FPGAs. These FPGAs can be configured by specifying the hardware description in an HDL, which means they can be reconfigured as needed [8]. This feature makes FPGA more useful. Finding the epoch locations is difficult task due to time varying nature of vocal tract system and excitation source [9]. However, the challenge arises when translating them into hardware descriptions (such as VHDL or Verilog) for FPGAs, which can be a non-trivial task because FPGAs have limited resources compared to software-based platforms. Ensuring that the algorithm fits within the available logic elements.

In this chapter we first discuss some of the techniques of epoch extraction that exist in literature. Next, we discuss the literature method which we use as baseline method, then the algorithm [4] which we use for HDL based implementation. The method which is proposed using HDL achieves stability as it achieves on software. Owing to the benefits due to lower complexity, the algorithm can be useful in many real time voice-based applications.

2.2 State of the art epoch extraction method

Epoch is the instant at which significant excitation of vocal tract filter takes place during the production of speech phonation. Epoch locations derived from speech signal are used in many applications such as prosody modification [10], glottal inverse filtering [11], [12], text-to-speech synthesis [13], glottal source analysis [14], [15] emotional speech analysis [16], telephonic speech analysis [17] and pathological speech analysis [18]. EGG is a method which measures vocal fold contact without affecting the speech production [19], [20], [21]. The EGG device tracks changes in resistance to a tiny electric current that goes between two sensors placed on the neck. These changes happen as the area

of contact of the vocal folds changes during voicing. The vocal fold contacting is monitored through the measurement of differences in this current flow [22]. During the voiced speech, EGG signal shows quasi-periodicity as vocal fold vibrates. Thus, when vocal fold are in contact, a larger signal is produced.

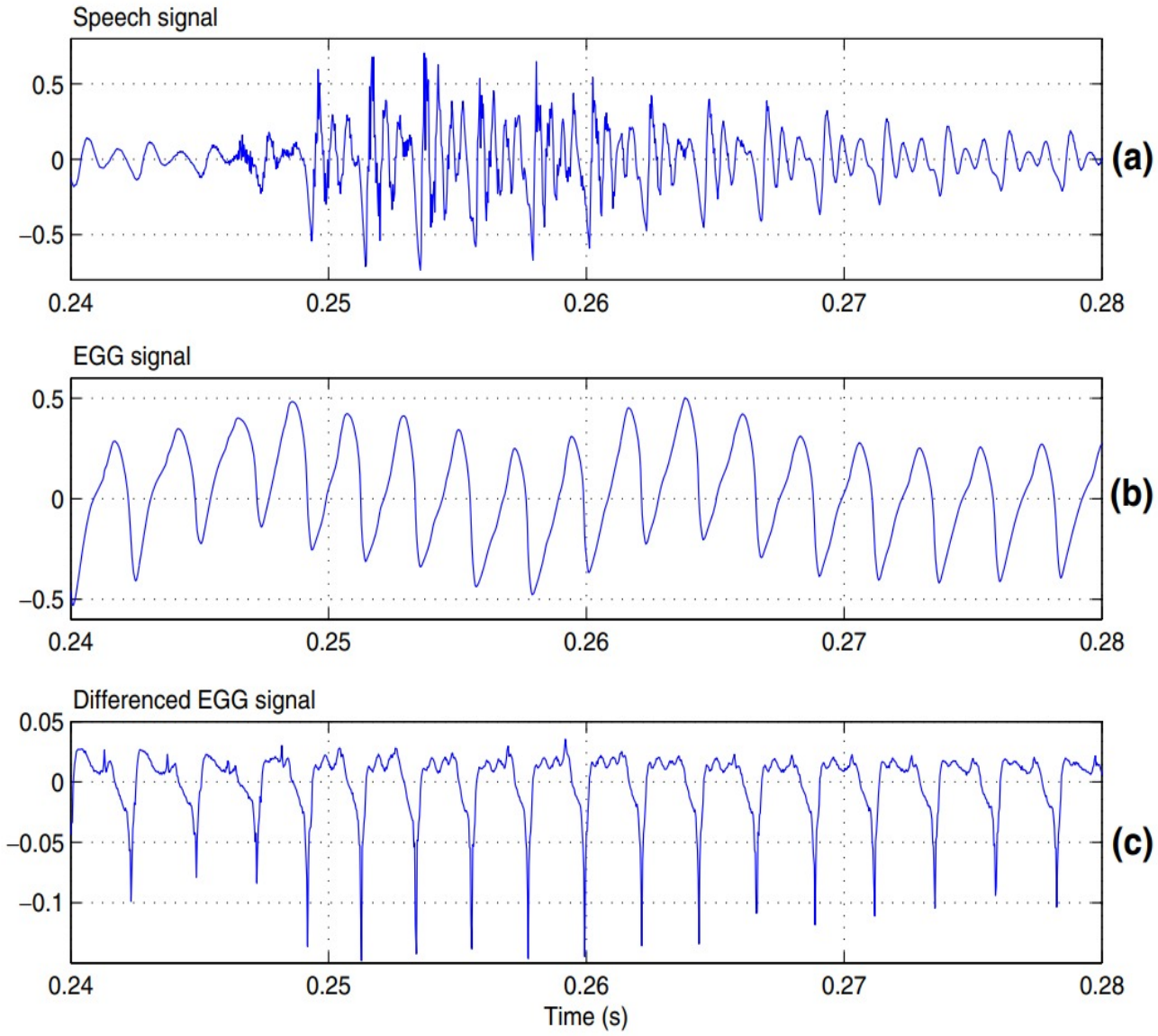

Figure 2.1 Extraction of epoch locations from differenced EGG signal. (a) A segment of speech signal, (b) EGG signal, (c) Differenced EGG signal.

Figure.2.1 from [9] shows that the location of negative peaks in differenced EGG signal refers to GCI locations. It can be noticed from Figure.2.1 that, the EGG signal is hardly affected by time varying vocal tract system. Therefore, the epoch locations and their strengths can be accurately determined from

the EGG signal even in the dynamic regions where the vocal-tract system is not stationary. However, the multiple dataset are not available for EGG system and also it is not usually availabe in practice. Hence, many speech processing algorithms have been used in the literature to derive the GCI locations such as, dynamic programming phase-slope algorithm (DYPSA) [23], yet another GCI/GOI algorithm (YAGA) [24], glottal closure/opening instant estimation forward backward algorithm (GEFBA) [25], zero frequency filtering (ZFF) [3], and zero-phase zero frequency resonator (ZP-ZFR) [4]. Precise detection of epoch locations is difficult task due to time varying nature of vocal tract system and excitation source [9].

2.2.1 DYPSA Method

The dynamic programming phase slope algorithm (DYPSA) utilizes excitation source information to detect GCIs [23]. The epoch detection procedure involves three steps: group delay function calculation, phase slope projection, and dynamic programming. First, the group delay function is derived from the linear prediction (LP) residual signal. Next, GCIs are estimated based on negative zero crossings of the group delay function. Subsequently, phase slope projection is employed to identify GCIs that were missed by the group delay function. While these steps capture most of the true GCIs, they also produce false GCIs. Therefore, the dynamic programming algorithm is applied to select the true GCIs by minimizing various cost functions.

2.2.2 YAGA Method

The Yet Another GCI Algorithm (YAGA) [24] utilizes the excitation signal (glottal flow waveform) to locate epochs. It combines various approaches used in other GCI detection methods, such as wavelet analysis, group delay function, and M-best dynamic programming. To highlight discontinuities in the glottal flow waveform, the multi-scale product of the stationary wavelet transform is employed, and then negative-going zero crossings of the group delay function are used to detect discontinuities. Falsely detected candidate GCIs are then removed using an M-best dynamic programming approach.

2.2.3 GEFBA Method

The Glottal Closure/Opening Instant Estimation Using Forward-Backward Algorithm (GEFBA) utilizes a source signal extracted through linear prediction-based inverse filtering [25]. This algorithm operates in two phases. The first phase involves deriving the glottal flow derivative from inverse filtering based on LP analysis. Subsequently, the second phase of the GEFBA algorithm employs a forward and backward algorithm to estimate GCIs for each voiced frame [18].

2.2.4 ZFF Method

ZFF is one of the best and simple technique to find the epoch locations in both degraded and clean speech signal. In this method, we pass the speech signal through zero frequency resonator to attenuate

the high frequency components which occur due to vocal tract system. The advantage of this resonator is that the characteristics of vocal tract system will not effect the characteristics of the discontinuities in the output of the resonator at (0 Hz). This is because the vocal-tract system has resonances at much higher frequencies than at the zero-frequency [9], which then highlights the excitation source characteristics. The ZFF method entails passing the differenced speech signal through a cascade of two ideal zero-frequency resonators. This approach is a narrow band filter with poles located at the unit circle, thus the resulting signal exhibits an approximate growth/decay pattern following a polynomial function of time. The zero-frequency filtered signal, obtained by removing the trend component, exhibits negative-to-positive zero crossings at instants corresponding to the epochs for positive polarity speech signal [26], [27].

2.2.5 ZP-ZFR Method

ZP-ZFR is one of the best, simple and stable epoch detection technique, to find the epoch locations in both degraded and clean speech signal. This method is also known as stable ZFF method. The ZP-ZFR provides stabilty to estimate accurate epochs from speech signal. This method has been implemented recently to overcome instability problems. Many approaches have been proposed in the recent years to overcome instability, but most of them require a higher filter order and are complex in design. This method entails passing the differenced speech signal through ZP-ZFR resonator and then perform the trend removal operation on output of ZP-ZFR resonator signal. It is a stable method that provides better performance than ZFF, due to the location of poles used inside the unit circle. In this method, negative peaks in trend removal signal correspond to the epoch locations of a speech signal [4].

2.3 Significant Gap

In literature many speech processing algorithms have been used to derive the GCI locations as mentioned above. In the majority of the techniques mentioned previously, epochs are identified using block processing approaches, leading to uncertainty about the precise location of the epochs. The majority of current techniques depend on the LP residual signal, which is obtained by inverse filtering the speech signal. While most of the time these approaches are effective, but they need to deal with certain issues like, setting threshold value, selection of parameters etc [3]. However, ZFF technique has been known to be robust in the estimation of epoch locations. The excellence of the zero frequency filtering lies in its simple technique. Although the ZFF has four poles on unit circles, the repeated poles on the unit circle make the ZFF unstable. So, the response of the system may grow or decay rapidly, which makes system marginally stable. Therefore, the stable ZFF technique is proposed which is known as ZP-ZFR method [4], to address the stability problem. However, since the stable ZFF has not yet implemented using FPGA, we implemented this approach using HDL for accurate detection of epochs on FPGA.

2.4 Summary and Conclusion

Epoch locations are important parameter in speech signals. An epoch occurs around the glottal closure and is also known as glottal closure instants, which are fundamental building blocks of voiced speech. Epochs play a important role in various aspects of speech processing and speech-related applications, which aims to capture the underlying mechanism that generates voiced speech. Epochs are essential for modeling the voice source, which is the underlying mechanism that generates voiced speech. Accurate determining epoch locations from the speech signal is a crucial step for various real-time applications.

In the literature, there are many research works that have been conducted to identify prominent epoch locations. Among these, ZFF stands out as one of the simplest and most effective techniques for locating epochs. It gives high identification rate than most of the epoch extraction methods. However, ZFF has four poles on unit circles and the repeated poles on the unit circle make the ZFF unstable. So, the response of the system may grow or decay rapidly which results in noisy and inaccurate output. To overcome this instability problem, various approaches has been proposed in previous research works. In the recent years one of the work is ZFF using FIR [7]. But these methods require higher filter order and they are complex in design. Later, the ZP-ZFR technique is proposed which is an IIR filter which requires lower filter order [4]. ZP-ZFR is a stable version of ZFF, which gives better performance than ZFF, due to the location of poles used in implementation of ZP-ZFR is taken inside the unit circle. As we know epochs has significant excitation points in speech signals, and it holds paramount importance in various real-time applications. Therefore epoch extraction methods, ZFF as a baseline method and ZP-ZFR is implemented on FPGA using HDL Verilog (Xilinx Vivado 2021.1). Nowadays, many digital electronic systems are implemented on FPGAs. These FPGAs can be configured by specifying the hardware description in an HDL, which means they can be reconfigured as needed.

The ZP-ZFR is an IIR filter which requires lower filter order. ZP-ZFR is a stable version of ZFF, which gives better performance than ZFF, due to the location of poles used in implementation of ZP-ZFR is taken inside the unit circle. Thus, high precision is not required in ZP-ZFR method. Instability in the system can introduce false alarms or misses in detecting epoch regions in speech signal. Therefore, stability is a crucial factor in speech systems, as an unstable system can lead to noisy and erroneous detections. Hence, stability ensures consistent and reliable performance. Thus, the primary goal is to implement a robust epoch detection method and its application, the ZP-ZFR based VAD, using HDL verilog.

Chapter 3

Stable Implementation of ZP-ZFR Algorithm using FPGA

This chapter focuses on the FPGA based implementation of epoch extraction method. Epochs, which has significant excitation points in speech signals, hold paramount importance in various real-time applications. The key challenge lies in accurately determining epoch locations from the speech signal, a crucial step for real-time applications. Therefore this section represents, recently proposed one of the useful epoch detection method ZP-ZFR [4] on FPGA and uses Zero Frequency Filtering as a baseline method [3].

3.1 ZP-ZFR algorithm and its HDL Implementation

The ZP-ZFR algorithm is a digital signal processing technique used to extract significant instants in speech signals. It is designed to attenuate the vocal tract resonances while enhancing the excitation source of the speech signal. The ZP-ZFR algorithm emphasizes the glottal activity, which is the vibration of the vocal folds responsible for generating the voiced portions of speech. By enhancing glottal activity, the filter aims to improve the accuracy of detecting voiced regions in a speech signal. This resonator highlights the low-frequency components of a signal. The zero frequency resonator boosts the lower frequency regions, making it particularly effective for capturing glottal vibrations. This method is recently proposed [4] which ensures stability.

3.1.1 Analysis of Zero Phase Zero Frequency Resonator

This section discuss the frequency domain analysis of the ZP-ZFR. From [4], the transfer function of $H_{ZP\_ZFR}(z)$ is given as:

$$H_{ZP\_ZFR}(z) = H(z)H(z^{-1}) \tag{3.1}$$

$$= \frac{z^2}{(1 - rz^{-1})^2(r - z^{-1})^2} \tag{3.2}$$

Now, the frequency response of this filter is given by,

$$H_{ZP\ \ ZFR}(e^{j\ !}) = H(e^{j\ !})H\ \ (e^{j\ !}) \tag{3.3}$$

$$= \frac{1}{(1\quad 2r\cos(\ !) + r^2)^2} \tag{3.4}$$

Therefore, this epoch extraction algorithm provides a stable response with no phase distortion. Furthermore, from the given equations, we know that the phase response of ZP-ZFR algorithm is zero and the group-delay response is equal to zero samples. Thus, the more detail of this algorithm analysis can be seen in [4].

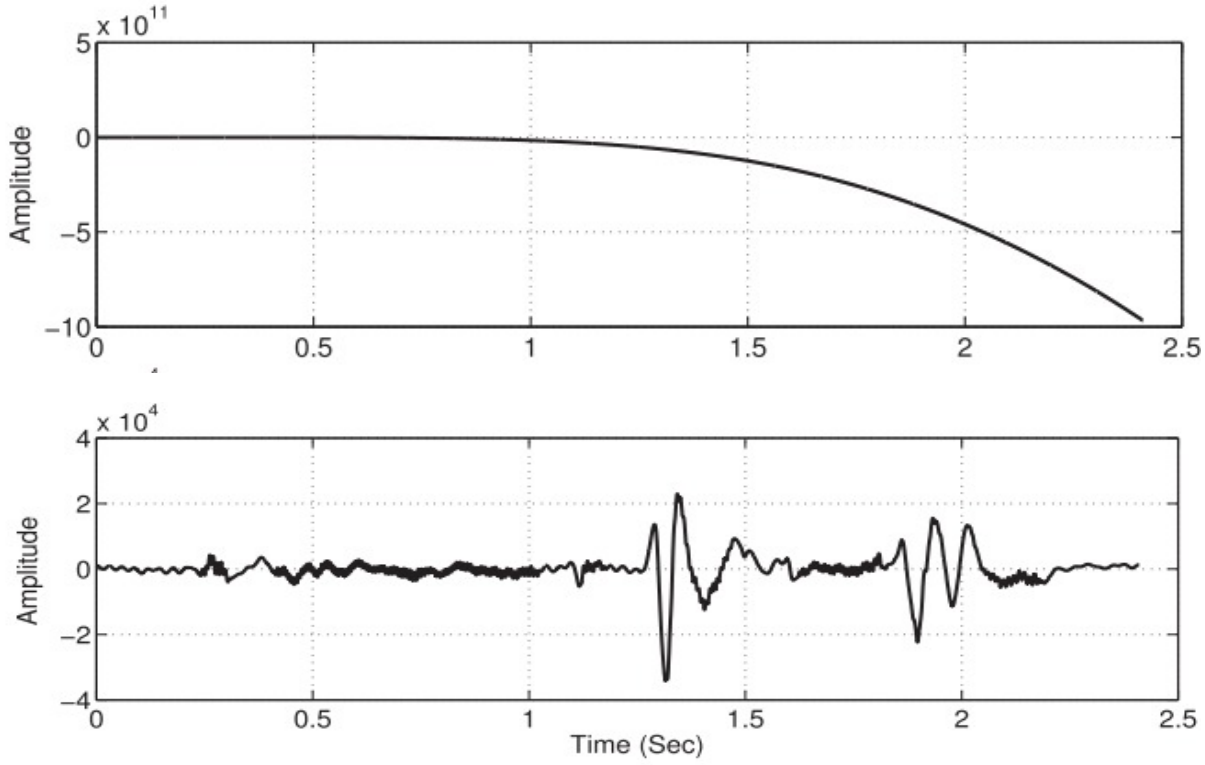

Figure 3.1 Response of ZFF, and ZP-ZFR filter at $r = 0.98$.

Figure.3.1 from [4], exhibits ZP-ZFR response at $r = 0.98$. The polynomial growth/decay can be observed in response of ZFF. The ZP-ZFR shows ocillations in its response. It can be seen that, in contrast to ZFF, polynomial growth/decay is not produced by the ZP-ZFR algorithm. The ZP-ZFR is an IIR filter approximation that requires lower filter order. The required filter order to meet a given specification is directly related to the hardware complexity, chip area, or computational speed of filter [28]. It is an effective and has simple (from design point of view) technique for finding accurate locations of epochs in clean and degraded speech signals. It was found from the study [4], this method provides

highest identification rate and lowest false alarm rate compared to other methods of epoch extraction. However, it is also to be note that the output of ZP-ZFR resonator demonstrates oscillations in its response. The oscillations in the resonator's response are correspond to very low frequency elements in speech. These low-frequency components are emphasized heavily due to the resonator's high gain around the zero frequency range. Unlike traditional ZFF methods, the response of the ZP-ZFR algorithm does not exhibit polynomial growth or decay. As a result, the ZP-ZFR method does not demand high precision for its operation [4].

Table 3.1 Filter characteristics of ZFF and ZP-ZFR

Filter	Stability	Phase	Causality
ZFF	unstable	linear-phase	causal
ZP-ZFR	stable	zero-phase	non-causal

Different properties of ZFF and ZP-ZFR are tabulated in Table.3.1. The ZP-ZFR system has a stable, non-causal, zero-phase response. Though the ZP-ZFR is a non-causal system, it can be efficiently used for epoch extraction from the pre-recorded speech signals.

3.1.2 Implementation of Zero Phase Zero Frequency Resonator

The complete block diagram representation of ZP-ZFR is shown in Figure.3.2. It consists of three sub-blocks named as, pre-emphasis block, zero phase zero frequency resonator block, trend removal or detrender block. The architecture has been designed as data flow graph for these sub-blocks which are modelled in HDL language Verilog. The steps involved in extracting the epoch locations from the speech signal using ZP-ZFR has been discussed below.

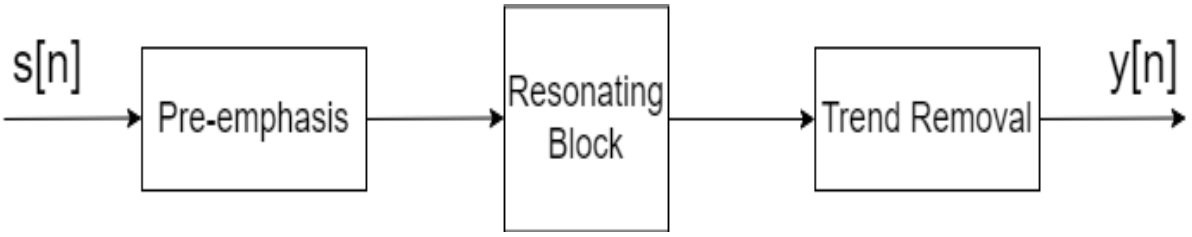

Figure 3.2 Block diagram representation of ZP-ZFR.

3.1.3 Pre-emphasis block of the ZP-ZFR

The given speech signal $s[n]$ passes through difference filter (pre-emphasis block) to remove low frequency fluctuations present in current signal.

$$x[n] = s[n] \quad s[n \quad 1] \tag{3.5}$$

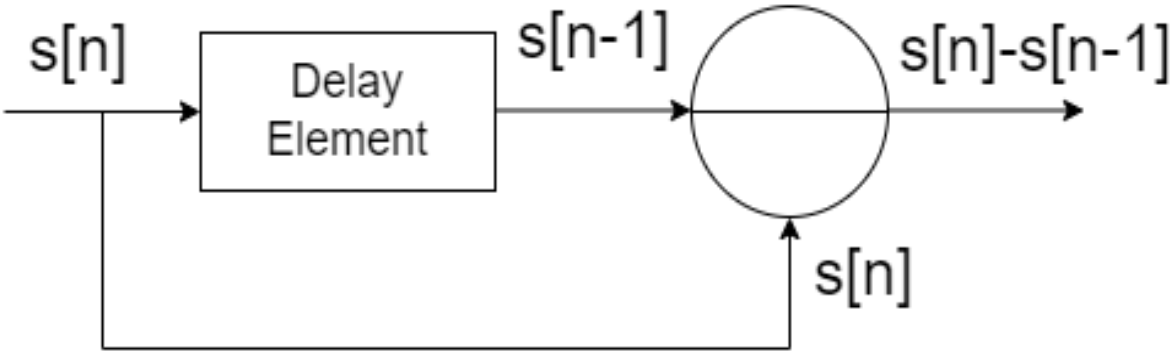

Figure 3.3 Data flow graph of Pre-emphasis block.

The data flow graph (DFG) in Figure.3.3 shows how Eq.3.5 is realised in hardware. From the Figure.3.3 we can notice that $s[n]$ is the speech signal and $s[n \quad 1]$ is delayed speech signal as it passes through delay element block.

The concept of pre-emphasis is analogous to audio equalization. The delay element is a fundamental part of this process, allowing for the comparison of the current signal with its immediate past. The resulting signal, after passing through the pre-emphasis block, is a key input to further stages in speech processing, including feature extraction and speech recognition algorithms.

3.1.4 Zero-Phase Zero Frequency Resonator block

After speech signal is processed through pre-emphasis block, we pass pre-emphasized speech signal to zero-phase zero frequency resonator block. The purpose of passing the speech signal through ZP-ZFR is to highlight the impulse like excitation and to reduce the effects of all high frequency resonances present around the signal due to vocal tract resonance.

The transfer function $H(z)$ of the filter is given as,

$$H_{ZP \quad ZFR}(z) = H(z)H(z^{\quad 1}) \tag{3.6}$$

Where $H(z)$ is given as,

$$H(z) = \frac{1}{(1 \quad rz^{\quad 1})^2} \tag{3.7}$$

and,

$$H(z^{\quad 1}) = \frac{1}{(1 \quad rz)^2} \tag{3.8}$$

From the above Eq.3.7 and Eq.3.8, we can observe that the system has poles at $z = r$ and $z = 1/r$ which denotes the pole position on the z-plane. The value of r determines the bandwidth of the resonator. For $0 < r < 1$, ZP-ZFR has magnitude response equivalent to the magnitude response of ZFF, and it also has zero phase and group-delay responses.

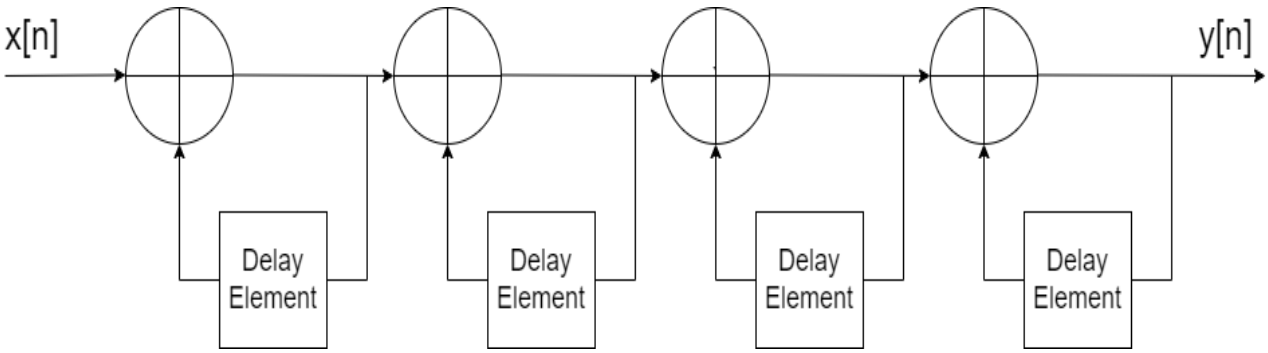

Figure 3.4 Data flow graph of resonator block.

The DFG shows in Figure.3.4 depicts the hardware realisation of Eq.3.9. The time domain response of ZP-ZFR is given by,

$$y[n] = a_1x[n] + a_2y[n \quad 1] \quad a_3y[n \quad 2] + a_4y[n \quad 3] \quad a_5y[n \quad 4] \tag{3.9}$$

In Eq.3.9 a_k are coefficients, $x[n]$ is a difference signal, $y[n]$ is the output of time-domain signal and $y[n \quad k]$ are delay elements. The difference signal $x(n)$ is passed through cascaded zero frequency resonator. The output signal of the cascade resonator is in stable state, as the filter contains poles inside the unit circle by taking coefficient value less than 1, i.e., 0:98 in present work. For the stability, the region of convergence (ROC) of the ZP-ZFR should include the unit circle. So the ROC of ZP-ZFR turns out to be an annular shape. By including the unit circle within it makes the whole system stable and non-causal. Though the ZP-ZFR is a non-causal system, it can be used for epoch extraction from the pre-recorded speech signals.

The bandwidth of the resonator in ZP-ZFR is determined by the choice of 'r'. The lower 'r' values result in a higher bandwidth for ZP-ZFR. Therefore, it leads to the presence of higher order harmonics in the ZP-ZFR response, which makes it difficult in finding the epoch locations due to the increased false alarms. For different values of r (0.8, 0.85, 0.90, 0.95, 0.96, 0.97, 0.98, 0.99), the performance of ZP-ZFR is evaluated. It was observed that when 'r' values ranged between 0.95 and 0.99, the performance of ZP-ZFR is equivalent to ZFF. However, for 'r' values less than or equal to 0.9, the performance of ZP-ZFR is degraded due to a rise in false alarms. Here from [4] we know that, the responses of ZP-ZFR do not produce polynomial growth/decay, which is common in ZFF method. Hence, high precision is not required in ZP-ZFR method. However, due to the significant gain around the zero frequency,

the speech harmonics are smeared over very low-frequency component. Therefore, the ZP-ZFR-based method for epoch extraction also requires the trend removal operation.

3.1.5 Trend removal block of the ZP-ZFR

The resonator has high gain around the zero frequency which smears harmonics of speech signal. This process helps in highlighting the discontinuities in the filtered signal due to impulse type of excitation. The value of window length used for trend removal operation ($2N + 1$ corresponds to the number of samples in the 15ms interval greater than 10ms to avoid false alarm) is not critical as long as it lies in the range of one to two pitch periods.

$$y[n] = y[n] \quad \frac{1}{(2N + 1)} \sum_{m=\ N}^{2N} y[n \quad m] \tag{3.10}$$

The Eq.3.10 represents as the detrender output signal depends on future samples which make it non-causal. So, for the real time implementation the output of trend removal is delayed by N samples to make causal as follows.

$$y[n \quad N] = y[n \quad N] \quad \frac{1}{(2N + 1)} \sum_{m=0}^{2N} y[n \quad m] \tag{3.11}$$

The DFG of Eq.3.11 is shown in figure.3.5 with latency of N clock cycles. In the current Hardware implementation as well, we have taken the value of N which is considered as 120 samples.

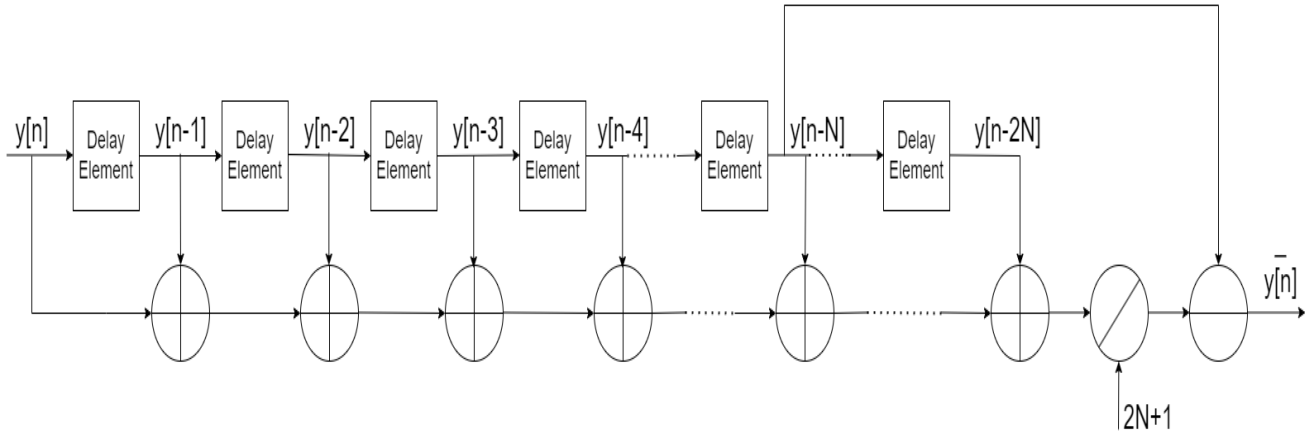

Figure 3.5 Data flow graph of trend removal block.

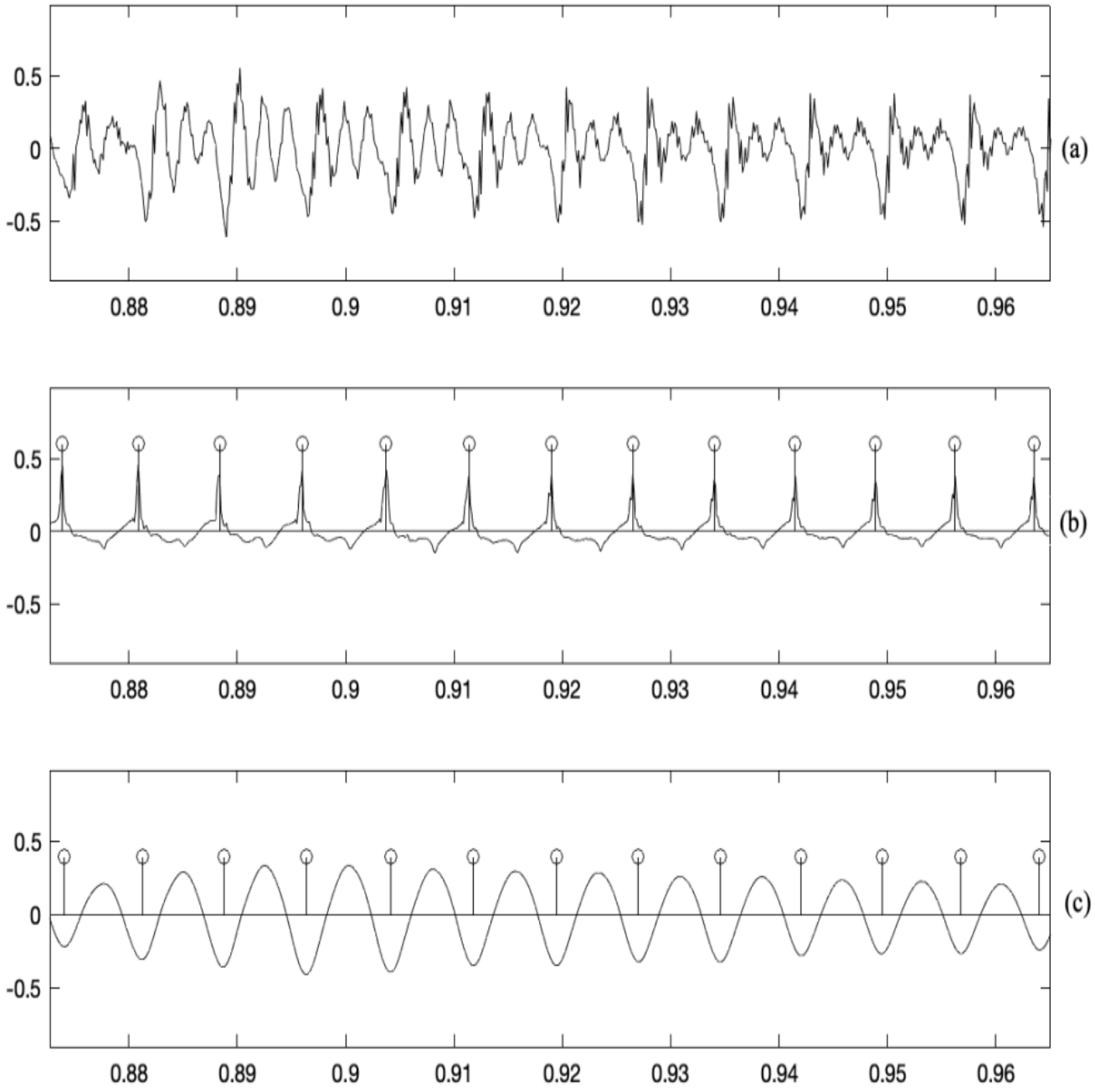

Figure 3.6 Illustrates the epoch detection using ZP-ZFR (a) A segment of speech signal taken from CMU-Arctic database, (b) part of EGG signal taken from database, (c) Hardware debugged zero phase-zero frequency resonator signal. Here we can see that the signal's epoch locations correlate to negative peaks for ZP-ZFR.

The Figure.3.6, depicts the epoch detection using ZP-ZFR algorithm. In the above figure.3.6, negative peaks in trend removal signal correspond to the epoch locations of a speech signal. Extraction of epoch locations from speech signal using the ZP-ZFR technique is demonstrated in figure.3.6.

3.2 Design flow using HDL

In this part design flow has been discussed. A simplified version of flow is shown in Fig.3.7.

3.2.1 Design Entry

Design has been entered in the system through different ways like Hardware Description Language. In this work we are working with HDL based design. HDL serves as the foundational framework for translating high-level design concepts into a language that hardware can understand. For our work, we have adopted an HDL-based design approach, specifically utilizing Verilog. This choice was driven by the need for a robust and well-established language in the field of hardware design. Verilog offers a structured and efficient means of describing the intricate interactions of our hardware components. It enables us to express complex logic, interconnections, and system behavior in a manner that facilitates testing and validation. The use of Verilog simplifies the translation of our design concepts into practical hardware elements, ensuring that our solutions are not only functional but also highly reliable.

3.2.2 Synthesis

The synthesis process produces a netlist of the design that can be utilized for further process. The synthesis verifies the correctness of the code syntax. It also examines the design's hierarchy and the accuracy of the article. It generates a netlist file and stores it as an NGC (Native Generic Circuit).

3.2.3 Implementation

In the typical design flow, verilog implementation is carried out after the design has been synthesized. During synthesis, the high-level description is transformed into a netlist, which represents the logical structure of the design. This netlist is then used for verilog implementation.

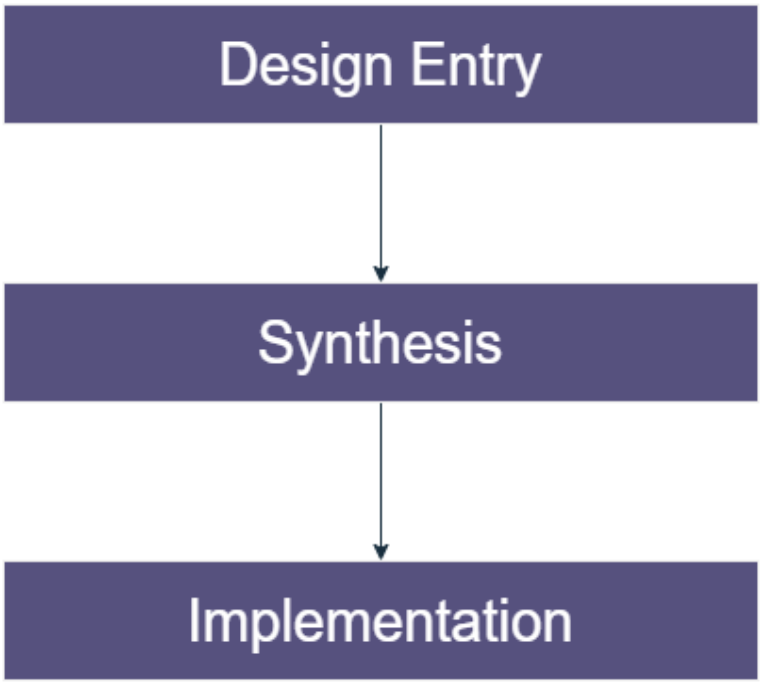

Figure 3.7 Design flow.

3.3 Experimental Setup

This section briefly explains the implementation details of the baseline method and displays the dataset.

3.3.1 Dataset

The database considered in our study for verilog implementation of ZP-ZFR using HDL is CMU Arctic database [29]. The reference locations of epochs from database are compared with the output of the ZP-ZFR. The database contains simultaneous recording of speech signal and corresponding EGG signal. All the speech signals are recorded at the sampling rate at 32kHz, later the speech signal was down sample to 16KHz. There are recordings of only English sentences in the given database. It contains three speakers, two male and one female speaker. All sentences recorded by the speakers contains around 1132 recordings of English sentence. The duration of each utterance is around 3 seconds.

3.3.2 Epoch extraction using base-line method

Epoch is the instant at which significant excitation of vocal tract filter takes place during the production of speech phonation. Zero Frequency Filtering is a simple and effective technique used to estimate the glottal closure instants accurately from the speech signal. The approach is based on narrow-band

filtering of the speech signal at zero frequency in order to gather evidence of epoch locations by reducing the impact of time-varying vocal tract resonances. However, the zero frequency filter is an unstable system. Hence, it may not be suitable for practical implementation due to the requirements of high precision computation.

3.3.2.1 Analysis of Zero Frequency Filtering

This section discusses the frequency domain analysis of the ZFF. From [3], ZFF is a causal and IIR system having the transfer function $H_{ZFF}(z)$, and it is given by,

$$H_{ZFF}(z) = \frac{1}{(1 - z^{-1})^4} \tag{3.12}$$

The ZFF has four poles on the unit circle (r = 1). From [4], the frequency response, magnitude response and phase response of the ZFF is given as following,

1. The frequency response of the ZFF is represented as,

$$H_{ZFF}(e^{j\omega}) = \frac{1}{(1 - e^{-j\omega})^4} \tag{3.13a}$$

$$= \frac{1}{(1 - \cos(\omega) + j\sin(\omega))^4} \tag{3.13b}$$

2. The magnitude response of ZFF is represented as,

$$|H_{ZFF}(\omega)| = \frac{1}{4(1 - \cos(\omega))^2} \tag{3.14a}$$

$$= \frac{1}{16\sin^4(\omega/2)} \tag{3.14b}$$

   From Equation.3.14a, it is observed that at $\omega = 0$, the magnitude response is unbounded. The ZFF has its maximum magnitude response at $\omega = 0$, and its magnitude response decays with increasing ω from 0 to $F_s/2$. Here, F_s corresponds to the sampling rate.

3. The phase response of the ZFF is represented as,

$$\phi_{ZFF}(\omega) = -4\tan^{-1}\left(\frac{\sin(\omega)}{1 - \cos(\omega)}\right) \tag{3.15a}$$

$$= 2(-\omega) \tag{3.15b}$$

From Equation.3.15b, it is observed that ZFF has linear phase response and constant group delay. We know it is concluded that, the ZFF is a causal, IIR filter having linear phase response and constant group-delay, i.e. no phase distortion [3]. However, the repeated poles on the unit circle make the ZFF unstable. So, the response of the system may grow/decay rapidly.

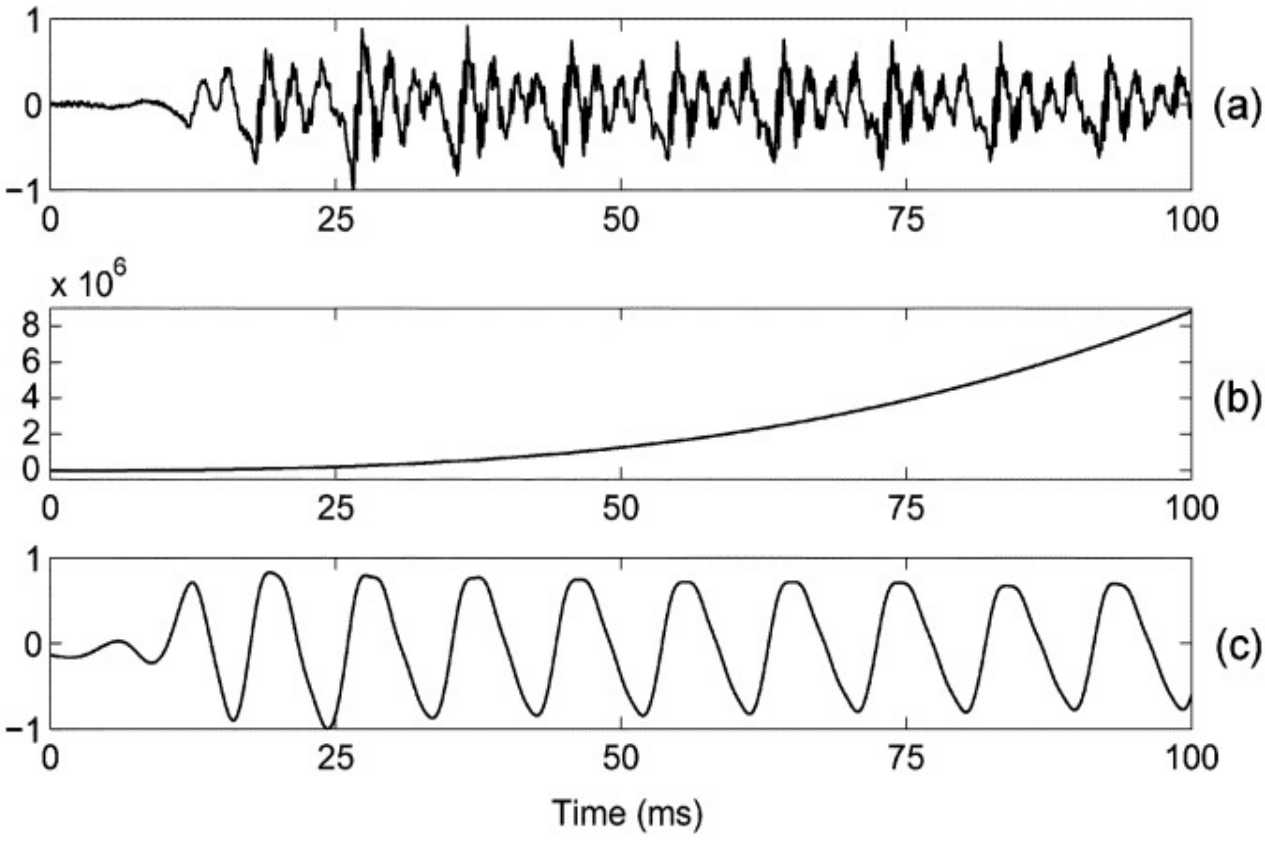

Figure 3.8 Illustration ZFF output. (a) speech signal, (b) output of resonators, and (c) mean subtracted signal.

Figure.3.8 from [3], illustrates speech segment, filtered output and its trend removal output signal. However this algorithm has very good identification rate of epochs and has simple design. This algorithm has implemented on FPGA in one of the literature [6]. But the output of the resonator is a exponentially growing / decaying function of time. The precision and the number of bits required to store this polynomial output increases with the length of the signal. So, the IIR filter is marginally stable. Due to the finite word length effects of digital filters, ZFF response may be stuck at saturation values. Consequently, the system may not be suitable in practice.

As to get stability, a lot of research have been done in several years. In the literature, stable version of ZFF was already implemented on FPGA. Some of the previous works are FIR implementations of zero frequency filter [6]. However, this method requires higher filter order [7], complex architecture, having more output delay and inferior performance than IIR filters. Therefore we have implemented Zero Phase Zero Frequency Resonator (ZP-ZFR) for epoch detection using Hardware Description Language(HDL), which is an IIR filter which requires lower filter order. ZP-ZFR is a stable version of ZFF. In the present work, ZFF and stable ZFF is implemented using Vivado 2021.1.

3.3.3 Performance measure

The system was replicated on hardware using verilog HDL after the algorithm's functionality was verified in MATLAB. Each individual block of the algorithm was implemented and tested thoroughly, to guarantee its accuracy and functionality. Next, we combined all these individual components into a complete integrated system and put it to the test by feeding it various inputs. The testing procedure verified that the hardware system functioned correctly and give the output as the MATLAB simulation. To test the system, we used the Artix-7 FPGA evaluation platform using Vivado 2021.1. We fed the system with static data that was generated in MATLAB. This data was stored in a BRAM (block RAM) and then fed to the other modules of the system. To evaluate the performance of the ZFF and ZP-ZFR algorithms on hardware, we examined the utilization of various components, including LUTs (lookup tables), flip-flops, and DSP slices which are shown in Table.3.2. These components are fundamental building blocks of FPGA, and these play crucial roles in implementing various digital circuits. Each component serves a distinct purpose and contributes to the overall functionality of FPGA-based designs.

Flip-flop is a basic memory element in FPGA. A flip-flop can be constructed by digital logic gates, and it's primary function is use to hold data which is fed to it. The given data will be stored in the flip-flop till it get's changed. A LUT is composed of a multiplexer and memory elements in its basic form [30]. LUTs are primarily used to implement combinational logic circuits, which perform logic operations on multiple input signals to produce a single output. LUTs are configurable, allowing FPGAs to be reprogrammed for different applications. Now coming to DSP, these are dedicated blocks for arithmetic and logic operations in recent FPGAs. These are called digital signal processing (DSP) slices [30]. Each DSP slice can perform several arithmetic and logic operations. Therefore, DSP slices will be very effective in implementation.

3.4 Result and conclusion

The utilization of hardware components by these filters is shown in Table.3.2. The power and hardware components LUT, Flip Flop, DSP slices utilizes more in ZP-ZFR. In future work, the power and hardware utilization of ZP-ZFR can be optimized.

Table.3.2 provides a comprehensive comparison of the hardware utilization between the ZFF and ZP-ZFR algorithm. The utilization metrics of various hardware components, including LUT, Flip-Flop, DSP

Table 3.2 Hardware Utilization of Zero Phase Zero Frequency Resonator.

S.No	Design	LUT	Flip Flop	DSP slices	Power
1	ZFF	17232	16704	0	0.188W
2	ZP-ZFR	25807	31424	40	1.684W

slices, and power consumption, were examined to assess the efficiency and resource requirements of each approach. For the ZFF implementation, we can notice that it utilized approximately 17232 LUTs, 16704 Flip-flops, and zero DSP slices. These values indicate the specific hardware resources utilized by the ZFF algorithm. On the other hand, the ZP-ZFR implementation exhibited slightly higher hardware utilization. It employed 25807 LUTs, 31424 Flip-Flops, and 40 DSP slices. Upon comparing the two algorithms on hardware, it is evident that the ZP-ZFR exhibits more hardware utilization. Despite the higher hardware and power utilization, we chose to implement the ZP-ZFR algorithm on hardware due to its stability. Stability is a crucial factor in most of the speech systems, an unstable system can lead to noisy and erroneous detection's. The ZP-ZFR algorithm, with its modifications and emphasis on resonances of the vocal tract, offers enhanced stability compared to the ZFF method. Thus, this stability ensures reliable and accurate detection of epochs.

In the present work, ZFF and ZP-ZFR is implemented using hardware description language verilog for identification of epoch locations from the given speech signal. The ZFF is simple and most accurate technique for finding epoch locations among most of the other algorithms, but it is marginally stable. The ZP-ZFR is stable implementation of ZFF with simple design and has better results than most of the ZFF parameters. But the power and hardware utilization of ZP-ZFR costs more than the ZFF method. ZP-ZFR's power and hardware consumption can be improved in future.

Chapter 4

Implementation of VAD using ZP-ZFR on FPGA

This chapter focuses on implementation of ZP-ZFR based application using HDL language (Verilog) on FPGA. In this work, a method for voice and unvoiced detection using epoch extraction is illustrate using ZP-ZFR. Voiced and unvoiced detection involves identifying the regions of speech when there is significant glottal activity and it holds paramount importance in various real-time applications or voice related technologies. Therefore this section represents voiced and unvoiced detection, based on one of the useful epoch detection method using HDL language (Verilog).

4.1 Voice-unvoice detection

Voice Activity Detection (VAD) is a widely used technique for detecting voiced and unvoiced segments. It involves identifying the regions of human speech when there is significant vibration of vocal folds. Such regions of speech are generally referred to as voiced regions or voiced speech. It is important to note that the term "voiced regions" refers to the parts of speech where the vocal cords vibrate strongly. This vibration creates a periodic sound wave that is characteristic of voiced speech sounds, and these vibrations don't have to be regular or periodic all the time. They can vary, such as in cases of strong aspiration or creaky voices. The non-voiced regions of speech include both silence (or background noise) and unvoiced speech. Unvoiced regions of speech include segments that do not involve the vibration of the vocal cords, such as sounds produced by voiceless fricatives (like "f" and "s") and stops (like "p" and "t").

4.2 Literature review on voiced-unvoiced detection

VAD is useful in various applications related to speech processing and communication systems. In previous literature, VAD are mostly based on the zero crossing rate (ZCR) [31], energy levels, formant shape [32], cepstral feature [33], adaptive modeling of voice signals [34], linear prediction coding (LPC) parameters [35]. However, these methods face challenges when it comes to setting accurate thresholds, which are critical in determining the performance of voiced/unvoiced detection. Additionally, most of these measures for determining voicing are sensitive to noise, and their effectiveness decreases as the

signal-to-noise ratio (SNR) decreases. There are also some frequently used methods such as Statistical models. These statistical models like neural network models, Gaussian mixture models (GMM), or hidden Markov models (HMM) are employed to combine information from various features [36], [37]. These methods do not depend critically on threshold setting, but require training data for different types of background noises. In general, these methods do not utilize the understanding of how speech is produced to a great extent. Additionally, most of these methods do not assess the ability to detect voiced and unvoiced regions of speech separately.

Recently authors, in the study [38] used ZFF algorithm for the VAD application. One of the key advantages of the ZFF is its robustness to noise. The ZFF algorithm depends on excitation source information [3] instead of speech signal for the detection of voice activity. However, the ZFF method is effective, but the response of the system may grow or decay rapidly which leads to the stability issue. To overcome the stability problem, ZP-ZFR algorithm is proposed in [4]. It is a stable version of ZFF. It has IIR filter which requires lower filter order. The stability of ZP-ZFR is due to the poles placed inside the unit circle. Hence, high precision is not required in ZP-ZFR method. This stability ensures consistent and reliable performance of the VAD system.

VAD is used in various applications such as speech coding, voice controlled systems. The speed, reliability, and real-time capabilities of hardware and FPGA implementations make VAD valuable in industries such as telecommunications, automotive, consumer electronics etc. In hardware implementation, unstable algorithms may yield unpredictable results. A stable implementation guarantees consistent and accurate identification of speech segments. Therefore we have implemented VAD using ZP-ZFR algorithm, because of its stability. A stable implementation guarantees consistent and accurate identification of speech segments.

4.3 ZP-ZFR Algorithm

The ZP-ZFR algorithm is a digital signal processing technique used to extract significant instants in speech signals. It is designed to attenuate the vocal tract resonances while enhancing the excitation source of the speech signal. The ZP-ZFR algorithm emphasizes the glottal activity, which is the vibration of the vocal folds responsible for generating the voiced portions of speech. By enhancing glottal activity, the filter aims to improve the accuracy of detecting voiced regions in a speech signal. This resonator accentuates the low-frequency components of a signal. The zero frequency resonator boosts the lower frequency regions, making it particularly effective for capturing glottal vibrations. By applying the ZP-ZFR filter to the speech signal, the voiced and unvoiced regions are effectively distinguished, allowing for better identification and analysis of the voiced segments. The ZP-ZFR filter have three following steps namely, pre-emphasis stage, resonator stage and trend removal stage.

In pre-emphasis stage, the speech signal passes through the difference filter to remove any time varying low frequency bias in the signal. Then, the output of the difference filter is passed through a resonator, which amplifies the frequencies associated with voiced segments. This filtering stage helps to

highlight the characteristics of the voiced segments in the signal. The ZP-ZFR filter output then requires a trend removal operation. This trend removal stage is used to eliminate trends or variations present in the speech signal. Trends in speech can arise due to factors such as variations in vocal tract shape, speaker-dependent characteristics, or environmental factors. By removing these trends, the algorithm can enhance the accuracy of subsequent processing steps and improve the detection and analysis of specific speech features.

4.3.1 Functionality of VAD based ZP-ZFR Algorithm

The ZP-ZFR algorithm plays a crucial role in accurately identifying voiced regions in a speech signal. It can enhance the performance of VAD algorithm by providing clearer cues for distinguishing between voiced and unvoiced sounds. VAD using the ZP-ZFR algorithm involves applying the ZP-ZFR technique to the complete speech signal. Then this ZP-ZFR output signal is split into frames to be processed which has overlapping window. In this work, the frame size we used is 30 ms and overlapping window of 20 ms. The energy calculation is applied to the short segments of the filtered output signal, which quantifies the strength or intensity of the signal. To determine the presence of human speech activity, the calculated energy of the output signal is compared with a predefined threshold. The ZP-ZFR filtered signal exhibits high energy in the voiced regions due to significant contribution from the impulse-like excitation as compared to the unvoiced regions of speech. If the energy exceeds the threshold value, it indicates the presence of voiced segments in the signal. Conversely, if the energy falls below the threshold value, it suggests the absence of voiced segment.

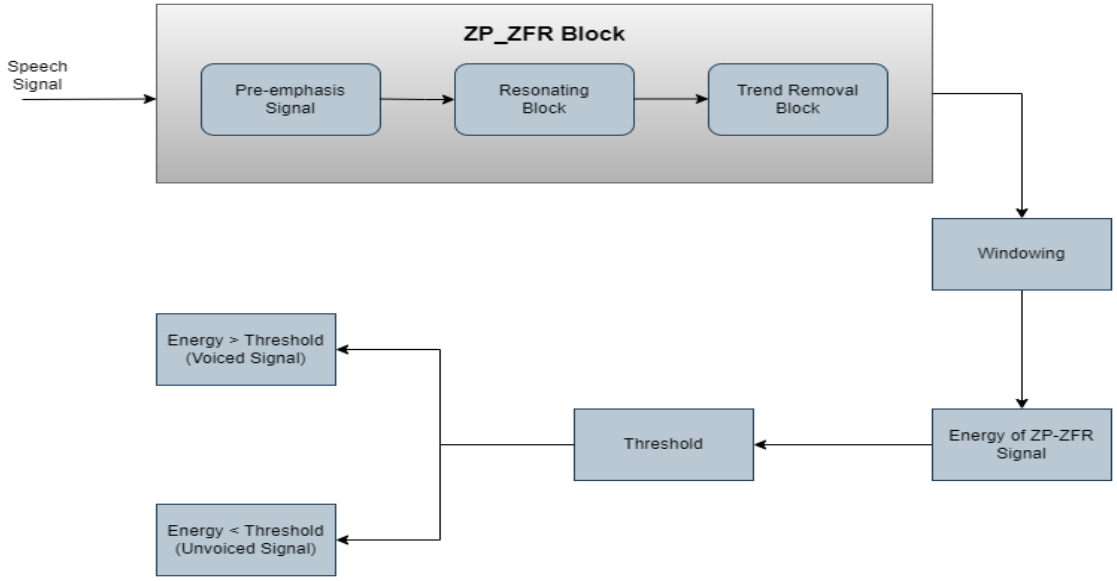

Figure 4.1 Block Diagram of ZP-ZFR-based Voice Activity Detection.

The Figure.4.1 depicts speech signal passes through pre-emphasis block then ZP-ZFR algorithm. Then this ZP-ZFR output signal is split into frames to be processed which has overlapping window. In this work, the frame size we used is 30ms and overlapping window of 20ms. The energy calculation is applied to the short segments of the filtered output signal, which quantifies the strength or intensity of the signal.

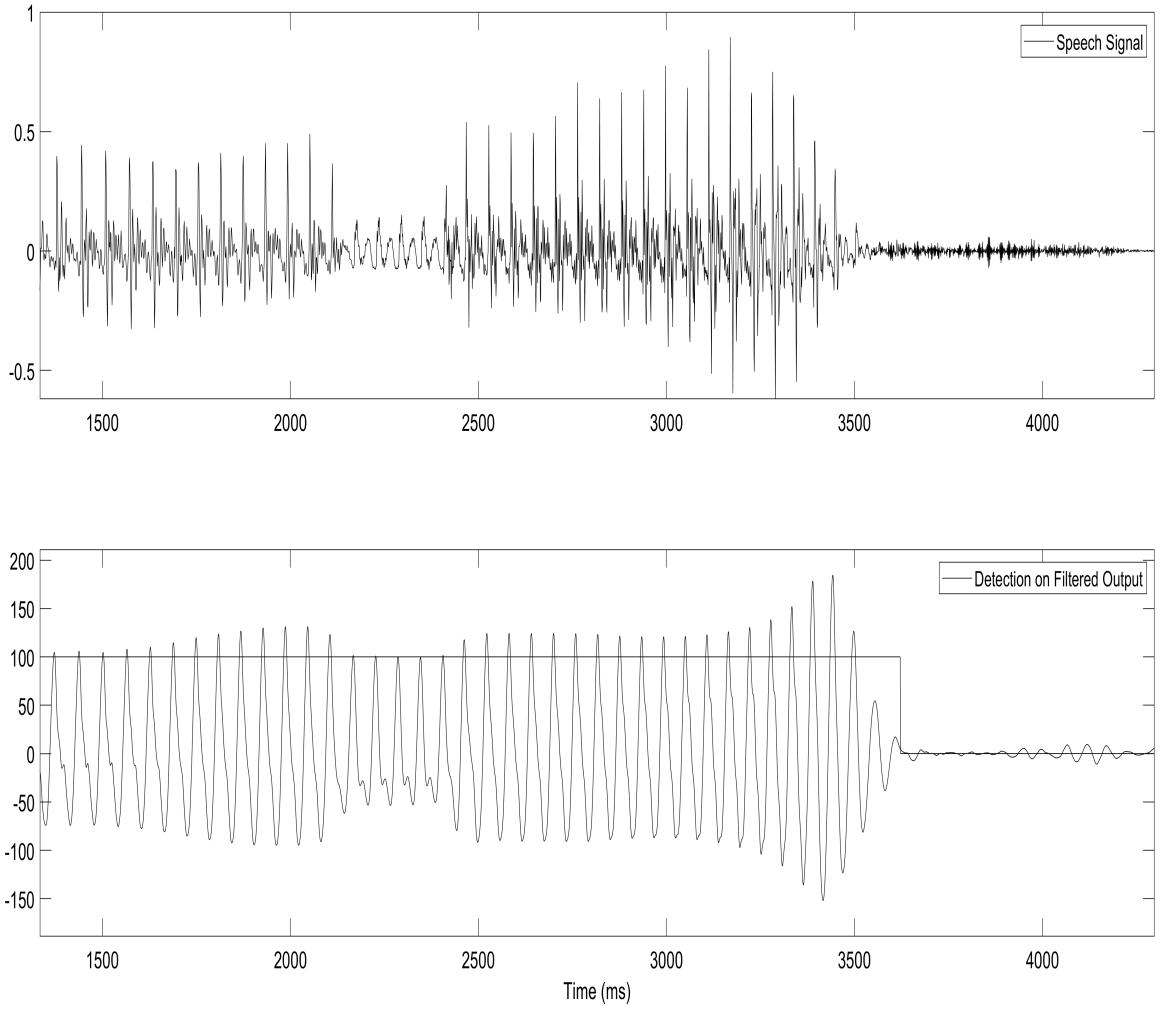

Figure 4.2 Voice Activity Detection using ZP-ZFR (a) A segment of speech signal (b) Voice Activity Detection using ZP-ZFR algorithm on segment of speech signal.

Figure.4.2 illustrates an example of the voice detection before its implementation using verilog. This is made using a signal composed of 3 ms. The decision of the ZP-ZFR based VAD is verified in MATLAB simulation. Figure.4.2 depicts speech segments were extracted from the TIMIT dataset, and these segments were used as input for the VAD using ZP-ZFR algorithm. From this, we can observe that the filtered output is correctly detecting most of the voice segments, as present in speech signal.

4.4 Implementation of ZP-ZFR-based VAD system using FPGA

The ZP-ZFR algorithm is designed to accurately detect voiced speech segments by analyzing the energy of the input signal within a specific frequency band. Implementing this algorithm on FPGA allows for real-time processing of audio signals with low latency. To implement VAD using ZP-ZFR, the algorithm is translated into a HDL such as verilog or VHDL. The HDL code specifies the logical operations, interconnections, and data flow required to perform VAD using ZP-ZFR. The algorithm can be divided into smaller computational units or modules, which can be implemented in parallel on the FPGA. The ZP-ZFR algorithm involves the calculation of energy levels and the comparison of these levels with a predefined threshold to determine the presence of speech activity. These calculations are performed using verilog, allowing for immediate detection of voice activity. Additionally, FPGA implementations offer flexibility and reconfigurability. The parameters of the ZP-ZFR algorithm, such as the location of poles and the threshold value, can be adjusted and optimized based on the specific requirements of the application. This adaptability allows for customization and optimization of the VAD system on the FPGA. FPGA-based implementations can provide lower power consumption compared to other computing platforms. FPGAs allow for efficient utilization of hardware resources, enabling power-efficient VAD systems that are suitable for battery-powered devices or applications with strict power constraints.

4.5 Experimental Setup

This section briefly provides the dataset used for VAD with Performance measure.

4.5.1 Dataset

The detection of voiced and unvoiced speech is evaluated on a subset of the TIMIT database. The subset consists of 38 speakers, 24 male and 14 female, uttering 10 short sentences each. All speech signals are recorded at a sampling rate of 16kHz. The database contains recordings of only English sentences. Performance is measured based on the algorithms used and a threshold value. A simple threshold on the excitation strength is used to detect the voiced epochs. Epochs derived from the clean speech using a ZP-ZFR, and the voiced/unvoiced decision derived from the manual markings, which are used to obtain the reference epochs in the voiced regions. An epoch in the voiced region is said to be missed if there is no epoch. The main source of error in the TIMIT dataset is manual marking. Two

kinds of errors can be introduced by manual labeling. First, the boundaries may not be very precise, and a few milliseconds of error are inevitable. Some weak voiced regions towards the vowel ending are typically overlooked. Also, the aspiration produced during some stop consonants tends to extend into the following vowel, making the boundary fuzzy. Second, manual errors are due to mismatch between speaker articulation and listener anticipation. Some sounds or regions that are susceptible to such errors are stop consonants and voiced fricatives.

4.5.2 Software evaluation Metrics

The accuracy of a VAD algorithm is typically measured using various performance metrics. These metrics provide a quantitative assessment of how well the VAD algorithm performs in detecting voice activity. This metric measures the overall correctness of the VAD system in detecting voiced and unvoiced segments in the speech signal. It can be calculated by comparing the VAD results with the ground truth annotations. The formula for calculating accuracy in VAD algorithms often involves the following terms:

- True Positives (TP): The number of correctly detected speech segments.
- False Positives (FP): The number of non-speech segments incorrectly classified as speech.
- False Negatives (FN): The number of speech segments incorrectly classified as non-speech.
- True Negatives (TN): The number of correctly detected non-speech segments.

Using these terms, the formula for calculating accuracy can be expressed as:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{4.1}$$

The accuracy metric provides an overall measure of how well the VAD algorithm performs in correctly classifying the voiced and unvoiced segments. A higher accuracy value indicates a more reliable VAD algorithm in accurately identifying voice activity. The accuracy is evaluated using appropriate metrics, which demonstrated the algorithm's effectiveness in identifying voiced regions in speech signals. In VAD the accuracy refers to the measure of how accurately the VAD system classifies segments of speech as either voiced or unvoiced. It provides an overall measure of the VAD system's ability to correctly identify voiced and unvoiced regions in a speech signal.

4.5.3 Hardware performance measure

In the present work, to assess the ZFF and ZP-ZFR based VAD applications on hardware we examine the utilization of various components. The hardware components LUT, Flip Flop, BRAM, DSP slices and power are shown in Table.4.2. These components are fundamental building blocks of FPGA, and these play crucial roles in implementing various digital circuits. Each component serves a distinct

purpose and contributes to the overall functionality of FPGA-based designs. The brief explanation about LUT, flip flop etc, is given in section.3.3.3.

Before proceeding with the hardware implementation of ZP-ZFR based VAD, we initially implemented and tested the algorithm using MATLAB software. This software implementation allowed us to verify the functionality and effectiveness of the ZP-ZFR method in detecting voice activity. Each individual block of the algorithm was implemented and tested its accuracy and functionality. Then we combined all these individual components into a complete integrated system and put it to the test by feeding it various inputs. The testing procedure verified that the hardware system functioned correctly and give the output as the MATLAB simulation. To test the system, we used the Artix-7 FPGA platform using Vivado 2021.1. We fed the system with static data that was generated in MATLAB.

In addition to ZP-ZFR, we also considered and implemented the ZFF based VAD method in our study. ZFF serves as a baseline method against which we compare the performance of ZP-ZFR. By including both methods, we aim to assess their respective capabilities in accurately detecting voice activity in speech signals. The ZP-ZFR method introduces additional modifications by including zero-phase, which refers to the property of maintaining phase linearity in the filtered signal. Therefore, it is important to note that ZP-ZFR is a modified and stable version of the ZFF.

4.6 Experimental results and conclusion

In this section, we illustrate the ZP-ZFR-based VAD output result in MATLAB. The speech utterance is taken from a dataset, then this speech signal is passed through the ZP-ZFR algorithm. Voice detection is then applied to the ZP-ZFR output signal, which is shown in Figure.4.2.

Table 4.1 Comparison of VAD Algorithm Performance.

S.No	Test case	Algorithm	Accuracy (in %)
1	Speech only	ZFF	94.7
2	Speech only	ZP-ZFR	94.9

In Table.4.1, the accuracy performance of ZP-ZFR based VAD is compared with the baseline ZFF based VAD method. In our comparative analysis of ZFF based VAD and ZP-ZFR based VAD, we examined the accuracy performance of both methods. The result presented in Table.4.1, reveal a slight difference in accuracy between the two approaches. According to the findings, the ZP-ZFR based VAD achieved a slightly higher accuracy rate of 94.9%, compared to the ZFF based VAD with an accuracy rate of 94.7%. The results indicate that integrating the ZP-ZFR modification into the VAD algorithm improves its capability to accurately distinguish between voiced and unvoiced segments in the speech signal. While the 0.2% does not make a big difference between two, however we choose this algorithm for implementation as the ROC of the ZP-ZFR include in the unit circle which makes system stable. The stability ensures consistent and reliable performance of the VAD system. In the context of VAD, an

unstable system can result in noisy and erroneous detection's of voice activity. Instability in the VAD system can introduce false alarms or misses in detecting voiced and unvoiced regions of speech. The false alarm rate indicates the rate at which the VAD system incorrectly detects non-voiced region as voiced segments. Similarly, the missed detection rate represents the rate at which the VAD system fails to detect actual voiced segments in the speech signal. Therefore, a lower missed detection rate indicates better performance and a lower false alarm rate signifies a more accurate system.

Now, the performance was compared to the hardware implementation of ZFF. The hardware implementation of ZP-ZFR maintained the accuracy and stability achieved in the software implementation, making it suitable for many applications that require low latency and high processing speed. Hence, the results of both the software and hardware implementation of ZP-ZFR demonstrated the algorithm's effectiveness in voice activity detection. Therefore, the software and hardware implementation provide a reliable and accurate solution, making it suited for various applications.

Table 4.2 Hardware Utilization Of Voice Activity Detector On FPGA.

S.No	Design	LUT	Flip Flop	BRAM	DSP	Power
1	ZFF VAD	32588	32085	59	0	1.219W
2	ZP-ZFR VAD	39321	32186	106	40	1.969W

Table.4.2 provides a comprehensive comparison of the hardware utilization between the ZFF based VAD and the ZP-ZFR based VAD. The utilization metrics of various hardware components, including LUT, Flip-flop, BRAM (Block RAM), DSP (Digital Signal Processor), and power consumption, were examined to assess the efficiency and resource requirements of each approach. For the ZFF based VAD implementation, we can notice that it utilized approximately 32588 LUTs, 32085 Flip-flops, 59 BRAM blocks, and zero DSP slices. These values indicate the specific hardware resources utilized by the ZFF based VAD algorithm. On the other hand, the ZP-ZFR based VAD implementation exhibited slightly higher hardware utilization. It employed 39321 LUTs, 32186 Flip-flops, 106 BRAM blocks, and 40 DSP slices. Upon comparing the two algorithms on hardware, it is evident that the ZP-ZFR based VAD exhibits more hardware utilization. Despite the higher hardware and power utilization, we chose to implement the ZP-ZFR based VAD algorithm on hardware due to its stability. From previous study, the ZP-ZFR on FPGA [39] have more hardware utilization than ZFF [6], [40], yet the ZP-ZFR has better stability and similar accuracy performance as ZFF. The hardware implementation of VAD using the ZP-ZFR algorithm on a FPGA offers a powerful and efficient solution for speech processing applications. However, stability is a crucial factor in voice activity detection systems as an unstable system can lead to noisy and erroneous detections of voice activity. The ZP-ZFR algorithm, with its modifications and emphasis on resonances of the vocal tract, offers enhanced stability compared to the ZFF method. This stability ensures reliable and accurate detection of voiced and unvoiced segments in speech signals.

In conclusion, the implementation of VAD using the ZFF and ZP-ZFR algorithms on FPGA using Xilinx Vivado 2021.1. The performance of both algorithms was evaluated on software and hardware,

showcasing their effectiveness in detecting voice activity in speech signals. The ZFF algorithm served as the baseline method for comparison, while the ZP-ZFR algorithm, as a modified version. The comparison between the two algorithms was conducted using the TIMIT database, providing a reliable dataset for evaluation. In software, the accuracy of both algorithms are almost same and in hardware, utilization of VAD based ZP-ZFR is more than ZFF based VAD. The reason to choose ZP-ZFR algorithm for VAD implementation is stability. In future work, the performance of the VAD system can be further enhanced by investigating alternative algorithms and refining the existing ones. Additionally, the hardware implementation can be optimized to achieve better resource utilization and scalability.

Chapter 5

Conclusion and Future Work

Initially this thesis describes implementation of epoch extraction and epoch based application using Hardware Description Language for real time applications. In the present work, ZFF and ZP-ZFR is implemented using verilog HDL for identification of epoch locations from the given speech signal. The ZFF algorithm serves as the baseline method for comparison with ZP-ZFR, as the ZP-ZFR algorithm is a stable implementation of ZFF. The CMU Arctic dataset was used to compare the two methods, offering a reliable dataset for evaluation. The ZFF is simple and most accurate technique for finding epoch locations among most of the other algorithms, but it is marginally stable. The ZP-ZFR is stable implementation of ZFF with simple design and has better results than most of the ZFF parameters. But the power and hardware utilization of ZP-ZFR costs more than the ZFF method. ZP-ZFR's power and hardware consumption can be improved in future.

However, another work presented which is the application based on ZP-ZFR algorithm. The implementation of VAD using the ZFF and ZP-ZFR algorithms on FPGA using Xilinx Vivado 2021.1. The performance of both algorithms was evaluated on software and hardware, showcasing their effectiveness in detecting voice activity in speech signals. The ZFF algorithm served as the baseline method for comparison, while the ZP-ZFR algorithm is a modified version of ZFF. The comparison between the two algorithms was conducted using the TIMIT database, providing a reliable dataset for evaluation. In software, the accuracy of both algorithms are almost same and in hardware, utilization of VAD based ZP-ZFR is more than ZFF based VAD. The reason to choose ZP-ZFR algorithm for VAD implementation is stability.

In future work, the performance of the VAD system can be further enhanced by investigating alternative algorithms and refining the existing ones. Additionally, the hardware implementation can be optimized to achieve better resource utilization and scalability.

Related Publications

- Syed Abdul Jabbar, Purva Sharma, Krishna Gurugubelli, Syed Azeemuddin and Anil Kumar Vuppala. “Implementation of Zero-Phase Zero Frequency Resonator Algorithm on FPGA” Proceedings of the 2022 Fourteenth International Conference on Contemporary Computing 2022, 24 OCT 2022
- Syed Abdul Jabbar, Purva Sharma, Krishna Gurugubelli, Syed Azeemuddin and Anil Kumar Vuppala. “Stable Implementation of Voice Activity Detector using Zero-Phase Zero Frequency Resonator on FPGA” 2023 IEEE International Conference and Expo on Real Time Communications at IIT (RTC). IEEE, 2023, Chicago USA, JULY 2023.

Bibliography

[1] G. Fant, "Acoustic theory of speech production (no. 2)," 1970.

[2] K. N. Stevens, *Acoustic phonetics*. MIT press, 2000, vol. 30.

[3] K. S. R. Murty and B. Yegnanarayana, "Epoch extraction from speech signals," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 8, pp. 1602–1613, 2008.

[4] K. Gurugubelli and A. K. Vuppala, "Stable implementation of zero frequency filtering of speech signals for efficient epoch extraction," *IEEE Signal Processing Letters*, vol. 26, no. 9, pp. 1310–1314, 2019.

[5] K. S. Srinivas and K. Prahallad, "An fir implementation of zero frequency filtering of speech signals," *IEEE transactions on audio, speech, and language processing*, vol. 20, no. 9, pp. 2613–2617, 2012.

[6] N. Srinivas, G. Pradhan, and P. K. Kumar, "Fpga implementation of zero frequency filter," in *2018 Conference on Information and Communication Technology (CICT)*. IEEE, 2018, pp. 1–5.

[7] P. Gangamohan and B. Yegnanarayana, "A robust and alternative approach to zero frequency filtering method for epoch extraction." in *INTERSPEECH*, 2017, pp. 2297–2300.

[8] G. Donzellini and D. Ponta, "A novel tool to introduce fpga in digital design laboratory," in *2012 9th International Conference on Remote Engineering and Virtual Instrumentation (REV)*. IEEE, 2012, pp. 1–8.

[9] B. Yegnanarayana and S. V. Gangashetty, "Epoch-based analysis of speech signals," *Sadhana*, vol. 36, pp. 651–697, 2011.

[10] K. S. Rao and B. Yegnanarayana, "Prosody modification using instants of significant excitation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 3, pp. 972–980, 2006.

[11] P. Alku, T. Murtola, J. Malinen, J. Kuortti, B. Story, M. Airaksinen, M. Salmi, E. Vilkman, and A. Geneid, "Openglot–an open environment for the evaluation of glottal inverse filtering," *Speech Communication*, vol. 107, pp. 38–47, 2019.

[12] M. Airaksinen, T. Raitio, B. Story, and P. Alku, "Quasi closed phase glottal inverse filtering analysis with weighted linear prediction," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 3, pp. 596–607, 2013.

[13] J. P. Cabral, K. Richmond, J. Yamagishi, and S. Renals, "Glottal spectral separation for speech synthesis," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 2, pp. 195–208, 2014.

[14] B. R. Gerratt, J. Kreiman, and M. Garellek, "Comparing measures of voice quality from sustained phonation and continuous speech," *Journal of Speech, Language, and Hearing Research*, vol. 59, no. 5, pp. 994–1001, 2016.

[15] P. Alku, "Glottal inverse filtering analysis of human voice production—a review of estimation and parameterization methods of the glottal excitation and their applications," *Sadhana*, vol. 36, no. 5, pp. 623–650, 2011.

[16] P. Gangamohan, S. R. Kadiri, and B. Yegnanarayana, "Analysis of emotional speech—a review," *Toward Robotic Socially Believable Behaving Systems-Volume I*, pp. 205–238, 2016.

[17] K. Gurugubelli, M. H. Javid, K. Alluri, and A. K. Vuppala, "Toward improving the performance of epoch extraction from telephonic speech," *Circuits, Systems, and Signal Processing*, vol. 40, no. 4, pp. 2050–2064, 2021.

[18] P. Barche, K. Gurugubelli, and A. K. Vuppala, "Comparative study of different epoch extraction methods for speech associated with voice disorders," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6923–6927.

[19] E. R. Abberton, D. M. Howard, and A. J. Fourcin, "Laryngographic assessment of normal voice: a tutorial," *Clinical Linguistics & Phonetics*, vol. 3, no. 3, pp. 281–296, 1989.

[20] B. Frøkjær-Jensen, "A photo-electric glottograph," *Annual Report of the Institute of Phonetics University of Copenhagen*, vol. 2, pp. 5–19, 1967.

[21] H. R. Gilbert, C. R. Potter, and R. Hoodin, "Laryngograph as a measure of vocal fold contact area," *Journal of Speech, Language, and Hearing Research*, vol. 27, no. 2, pp. 178–182, 1984.

[22] D. M. Howard, "Variation of electrolaryngographically derived closed quotient for trained and untrained adult female singers," *Journal of Voice*, vol. 9, no. 2, pp. 163–172, 1995.

[23] A. Kounoudes, P. A. Naylor, and M. Brookes, "The dypsa algorithm for estimation of glottal closure instants in voiced speech," in *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1. IEEE, 2002, pp. I–349.

[24] M. R. Thomas, J. Gudnason, and P. A. Naylor, "Estimation of glottal closing and opening instants in voiced speech using the yaga algorithm," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 82–91, 2011.

[25] A. I. Koutrouvelis, G. P. Kafentzis, N. D. Gaubitch, and R. Heusdens, "A fast method for high-resolution voiced/unvoiced detection and glottal closure/opening instant estimation of speech," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 2, pp. 316–328, 2015.

[26] S. R. Kadiri and B. Yegnanarayana, "Analysis of singing voice for epoch extraction using zero frequency filtering method," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4260–4264.

[27] T. Drugman, P. Alku, A. Alwan, and B. Yegnanarayana, "Glottal source processing: From analysis to applications," *Computer Speech & Language*, vol. 28, no. 5, pp. 1117–1138, 2014.

[28] A. V. Oppenheim, *Discrete-time signal processing*. Pearson Education India, 1999.

[29] J. Kominek and A. W. Black, "The cmu arctic speech databases," in *Fifth ISCA workshop on speech synthesis*, 2004.

[30] C. Ünsalan and B. Tar, *Digital system design with FPGA: implementation using Verilog and VHDL*. McGraw-Hill Education, 2017.

[31] J. Jean-Claude, "A study of endpoint detection algorithms in adverse conditions: Incidence on a dtw and hmm recognizer," *Eurospeach 1991*, 1991.

[32] J. D. Hoyt and H. Wechsler, "Detection of human speech in structured noise," in *Proceedings of ICASSP'94. IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 2. IEEE, 1994, pp. II–237.

[33] J. Haigh and J. Mason, "Robust voice activity detection using cepstral features," in *Proceedings of TENCON'93. IEEE Region 10 International Conference on Computers, Communications and Automation*, vol. 3. IEEE, 1993, pp. 321–324.

[34] N. Yoma, F. McInnes, and M. Jack, "Robust speech pulse detection using adaptive noise modelling," *Electronics letters*, vol. 32, no. 15, pp. 1350–1352, 1996.

[35] L. Rabiner and M. Sambur, "Voiced-unvoiced-silence detection using the itakura lpc distance measure," in *ICASSP'77. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2. IEEE, 1977, pp. 323–326.

[36] A. P. Lobo and P. C. Loizou, "Voiced/unvoiced speech discrimination in noise using gabor atomic decomposition," in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).*, vol. 1. IEEE, 2003, pp. I–I.

[37] B. Atal and L. Rabiner, "A pattern recognition approach to voiced-unvoiced-silence classification with applications to speech recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 3, pp. 201–212, 1976.

[38] N. Dhananjaya and B. Yegnanarayana, "Voiced/nonvoiced detection based on robustness of voiced epochs," *IEEE Signal Processing Letters*, vol. 17, no. 3, pp. 273–276, 2009.

[39] S. A. Jabbar, P. Sharma, K. Gurugubelli, S. Azeemuddin, and A. K. Vuppala, "Implementation of zero-phase zero frequency resonator algorithm on fpga," in *Proceedings of the 2022 Fourteenth International Conference on Contemporary Computing*, 2022, pp. 49–53.

[40] N. Srinivas, G. Pradhan, and D. Govind, "A simplified realization of zero frequency filter for hardware implementation," *Circuits, Systems, and Signal Processing*, vol. 39, pp. 4717–4729, 2020.