VLSI Design for Speech Feature Extraction Based on Levinson-Durbin Algorithm

A thesis submitted in partial fulfilment of the requirements for the award of the degree of

M.Tech.

in

Computer Science & Engineering (VLSI Design)

by

Nemana Udaya Ramya
(2014VLSI-12)

under the guidance of

Dr.Manisha Pattanaik

विश्वजीवनामृतं ज्ञानम्

ATAL BIHARI VAJPAYEE
INDIAN INSTITUTE OF INFORMATION
TECHNOLOGY AND MANAGEMENT
GWALIOR-474015

2016

Checklist

#	Items	Declaration	
1.	Is the thesis/report bound as specified?	Yes	No
2.	Is the Cover page in proper format as given in Annexure 1 of guidelines for thesis preparation?	Yes	No
3.	Is the Title page (inner cover page) in proper format?	Yes	No
4.	**I.** Is the Certificate from the supervisor in proper format? **II.** Has it been signed by the supervisor?	Yes	No
5.	**I.** Is the Abstract included in the thesis/report properly written within 400 to 600 words? **II.** Have the technical keywords (not more than six) specified properly?	Yes	No
6.	Have you included the List of Abbreviations/Acronyms in the thesis/report?	Yes	No
7.	Does the thesis/report contain a summary of the literature survey?	Yes	No
8.	Does the Table of Contents include page numbers? **I.** Are the pages numbered properly? (Chapter 1 should start on page number 1) **II.** Are the Figures numbered properly? (Figure Numbers and Figure Titles should be only at the bottom of the figures) **III.** Are the Tables numbered properly? (Table Numbers and Table Titles should be only at the top of the tables) **IV.** Are the Titles for the Figures and Tables proper and sources acknowledged? **V.** Are the Appendices numbered properly? Are their titles appropriate?	Yes	No
9.	Have you incorporated feedback received during various stages of evaluation?	Yes	No
10	Is the Conclusion of the thesis/report based on discussion of the work?	Yes	No
11	**I.** Are References or Bibliography given at the end of the thesis/report? **II.** Have the References been cited properly inside the text of the thesis/report? **III.** Is the citation of References in proper format?	Yes	No
12	Is the thesis/report format and contents are according to the guidelines?	Yes	No

Candidate Declaration

I hereby certify that I have properly checked and verified all the items as prescribed in the check-list and ensure that my thesis/report is in proper format as specified in the guideline for thesis preparation.

I also declare that the work containing in this report is my own work. I, understand that plagiarism is defined as any one or combination of the following:

1. To steal and pass off (the ideas or words of another) as one's own
2. To use (another's production) without crediting the source
3. To commit literary theft
4. To present as new and original an idea or product derived from an existing source.

I understand that plagiarism involves an intentional act by the plagiarist of using someone else's work/ideas completely/partially and claiming authorship/originality of the work/ideas. Verbatim copy as well as close resemblance to some else's work constitute plagiarism.

I have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programmes, experiments, results, websites, that are not my original contribution. I have used quotation marks to identify verbatim sentences and given credit to the original authors/sources.

I affirm that no portion of my work is plagiarized, and the experiments and results reported in the report/dissertation/thesis are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, I shall be fully responsible and answerable. My faculty supervisor(s) will not be responsible for the same.

Signature:
Name: Nemana Udaya Ramya
Roll No.:2014VLSI-012
Date:

Thesis Certificate

This is to certify that the thesis titled **VLSI Design for Speech Feature Extraction Based on Levinson-Durbin Algorithm**, in partial fulfilment of the requirement for the award of the Degree of Master of Technology and submitted by **Nemana Udaya Ramya (2014VLSI-12)**, to the ABV - Indian Institute of Information Technology & Management Gwalior, is an authentic record of the research work done by her during the period *July, 2015* to *May, 2016* under the supervision of **Dr.Manisha Pattanaik**. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma to best of my knowledge.

Date:

Signature of Supervisor
Name: Dr.Manisha Pattanaik
Designation: Associate Professor
ABV-IIITM Gwalior

Abstract

The thesis aims to develop a VLSI architecture for speech feature extraction that is based on Levinson Durbin algorithm. In this, speech feature extraction are expressed in terms of linear predictive coefficients and line spectral pairs. These modules are the sub blocks of the ITU recommended VOCODER G.729. These modules are the primary blocks in many speech coding algorithms. Since it is shown that software logic takes more algorithmic delay, need of hardware logic development was required. VOCODER G.729 is an ITU recommended voice codec that uses the principle of Algebraic Code Excited Linear Prediction(ACELP). ACELP is a type of voice codec that compresses the speech signal based on model parameters of human speech.

Firstly, to understand the functionality of the VOCODER G.729 MATLAB implementation is done. It is found that LD algorithm and LP-LSP conversion blocks take more algorithmic delay. Thus, corresponding hardware logic blocks are developed which speed up the operation. Area and power utilized by the architecture is found using Synopsys 90nm CMOS technology. The results show that hardware implementation takes less algorithmic delay than that of software logic. The thesis presents two methods of the hardware implementation for LD algorithm: one is a direct mapping of the algorithm and the other using pipeline method. Due to reduction in the hardware logic there is a 3.22 times reduction in area and 38.9 times decrease in power utilisation for implementing levinson durbin algorithm.

Keywords: LPC, LD algorithm, LSP, LSF, ACELP, Speech Coding, Codebook

Acknowledgement

I am highly indebted to **Dr.Manisha Pattanaik**, and obliged for giving me the autonomy of functioning and experimenting with ideas. I would like to take this opportunity to express my profound gratitude to her not only for her academic guidance but also for her personal interest in my report and constant support coupled with confidence boosting and motivating sessions which proved very fruitful and were instrumental in infusing self-assurance and trust within me. The nurturing and blossoming of the present work is mainly due to her valuable guidance, suggestions, astute judgment, constructive criticism and an eye for perfection. My mentor always answered myriad of my doubts with smiling graciousness and prodigious patience, never letting me feel that I am novices by always lending an ear to my views, appreciating and improving them and by giving me a free hand in my report. It's only because of her overwhelming interest and helpful attitude, the present work has attained the stage it has.

Finally, I am grateful to our Institution and colleagues whose constant encouragement served to renew my spirit, refocus my attention and energy and helped me in carrying out this work.

Date: **Nemana Udaya Ramya**

Contents

List of Figures

List of Tables

List of Abbreviations

LD	:	Levinson Durbin
LPC	:	Linear Predictive Coefficients
LSP	:	Line Spectral Pair
LSF	:	Line Spectral Frequency
PCM	:	Pulse Code Modulation
HDL	:	Hardware Description Language
VOCODER	:	Voice Coder Decoder
IC	:	Integrated Circuit
CS-ACELP	:	Conjugate Structure Algebraic Code Excited Linear Prediction
LP	:	Linear Prediction
MA	:	Moving Average
AR	:	Auto Regressive
VoIP	:	Voice Over Internet Protocol
VLSI	:	Very Large Scale Integration
RC	:	Reflection Coefficients
ACB	:	Adaptive Codebook
ICB	:	Innovation Codebook
FCB	:	Fixed Codebook
ISPP	:	Interleaved Single Pulse Permutation

Chapter 1

Introduction

ITU Recommendation G.729 is an 8 kbit/s CS-ACELP Speech Codec. It is an umbrella of vocoder standards. It is an adaptive digital technology that performs voice compression at bit rates that vary between 6.4 and 12.4 kbps.

G.729 uses a 16-bit PCM signal as input. Speech coding algorithm that is used in vocoder is Conjugate-Structure Algebraic-Code-Excited Linear Prediction (CS-ACELP. For each frame of 10 ms,the input speech is analysed and the CELP parameters are extracted, encoded and transmitted. The CELP parameters include LPC-filter coefficients, indexes of the codebooks (adaptive codebook and fixed codebook) and their respective gains.The decoder takes the received parameters , extracts the excitation signal from the fixed and adaptive codebook which is also known as long term synthesis filter and forms the short synthesis filter.

Pros and Cons of G.729 VOCODER: The advantages of VOCODER G.729 are it compresses the speech data as low as 10ms thus providing low delay. Because of this, music or tones such as DMF or fax tones are not been transported reliably with this codec. Since it has lower bandwidth around 8kbps it is mostly used in applications like Voice Over IP(VoIP) for its low bandwidth requirement. The disadvantages of this codec are license is required for use and decreases speech quality marginally.

1.1 Encoder

The input signal is first high pass filtered and then down scaled by a factor of two. This process is done in the pre-processing block. This pre processed signal is the input for the further analysis. The linear prediction procedure is followed once per 10ms frame that is 80 samples per frame. This procedure is done to find the LP filter coefficients. In order to do the analysis in frequency domain these LP coefficients are transformed into Line spectral pairs and line spectral frequencies. These coefficients are then quantized and interpolated using vector quatization having 18 bits in a prediction of two stage. Perceptually weighted distortion measure is used to minimize the error between original and the reconstructed speech signal. This method is known as analysis by synthesis method and the signal chosen by this method is the excitation signal. The improvement in the performance of input signal is achieved by making the perceptual weighing amount adaptive that is having a flat frequency response. The procedure of encoding is shown in Figure 1.1.

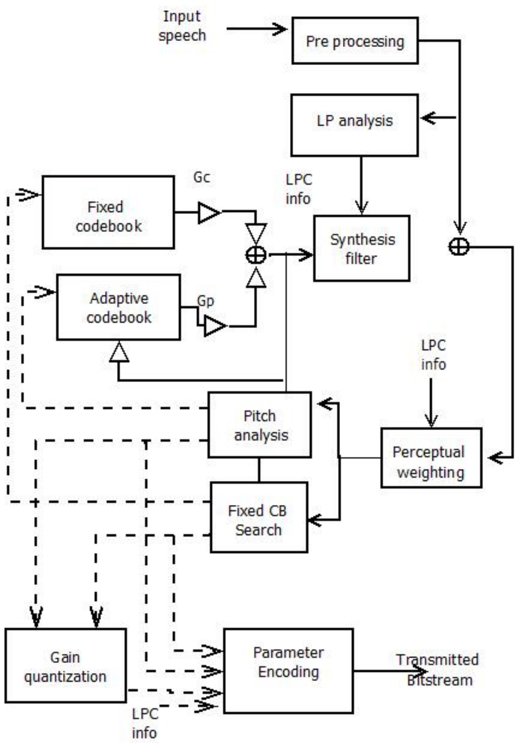


Figure 1.1: CS-ACELP Encoder: Encoding Principle [1]

The fixed codebook parameter and the adaptive codebook parameter which is collectively called as excitation parameters are determined for every 5ms subframe consisting of 40 samples. Quantized LP coefficients and unquantized LP coefficients that are interpolated are used in the first subframe while in the second subframe only quantized and unquantized filter coefficients are used. Based on the speech signal(perceptually weighted) an open-loop pitch delay is estimated for every 10ms frame. For each subframe the following operations are repeated. The signal flow representation is shown in Figure 1.2 and the corresponding data flow operation is shown in Figure 1.3.

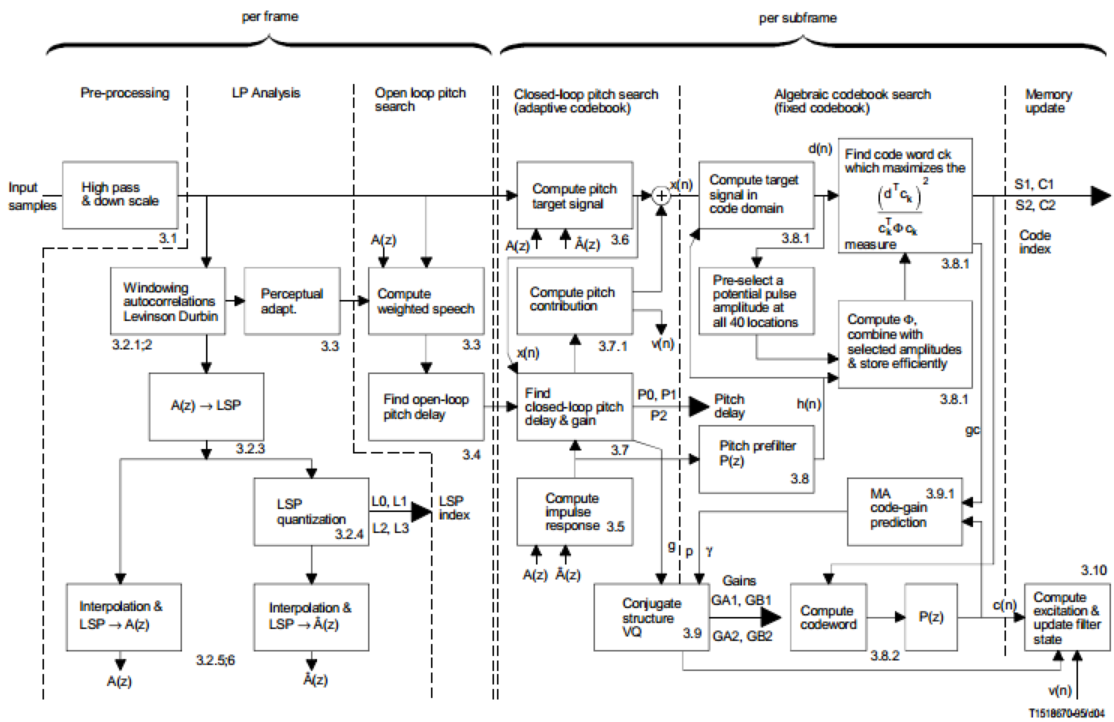


Figure 1.2: CS-ACELP Encoder: Signal Flow [1]

The weighted synthesis filter is used to compute the target signal x(n) which is done by the filtering the residual of Linear Prediction. Subtraction of zero input response of the weighted synthesis filter from the weighted signal is the common approach which is the phenomena used for updating the initial states of the filter. In this, the error between LP residual and excitation are filtered. After this procedure the impulse response h(n) of the weighted synthesis filter is calculated. To find the adaptive codebook delay and gain closed loop analysis is done.

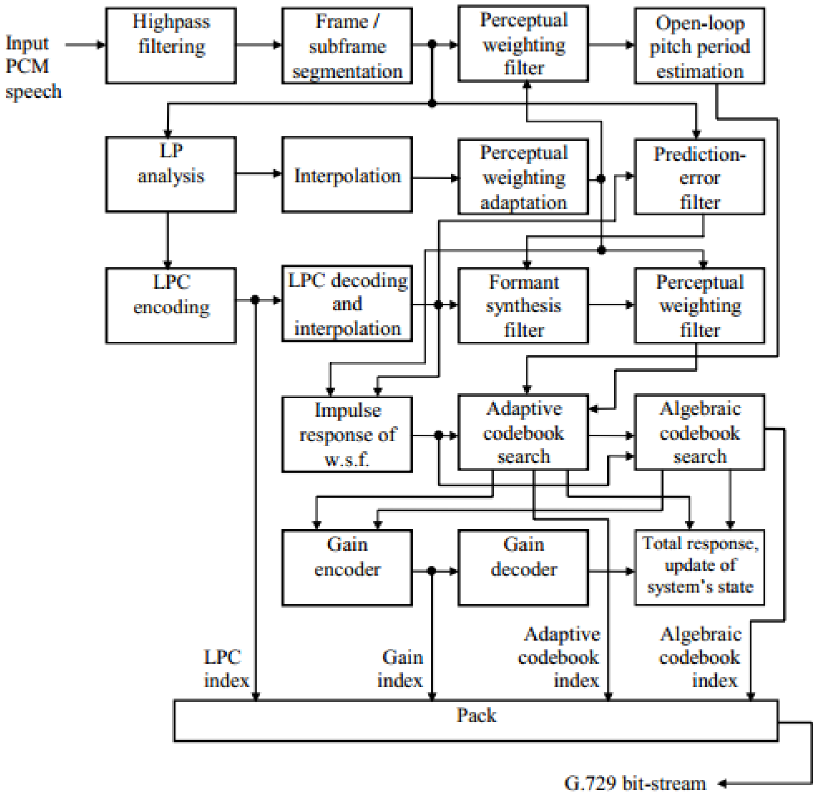


Figure 1.3: CS-ACELP Encoder: Data operation [12]

1.2 Decoder

The received bit-stream is used to extract the parameter's indices. For 10ms frame of speech, decoding is done for these indices. The parameters include: "LSP coefficients, the two fractional delays, the two fixed codebook vectors and the two sets of adaptive and fixed codebook gains". The decoding principle is shown in Figure 1.4 and the corresponding data operation is shown in Figure 1.5.

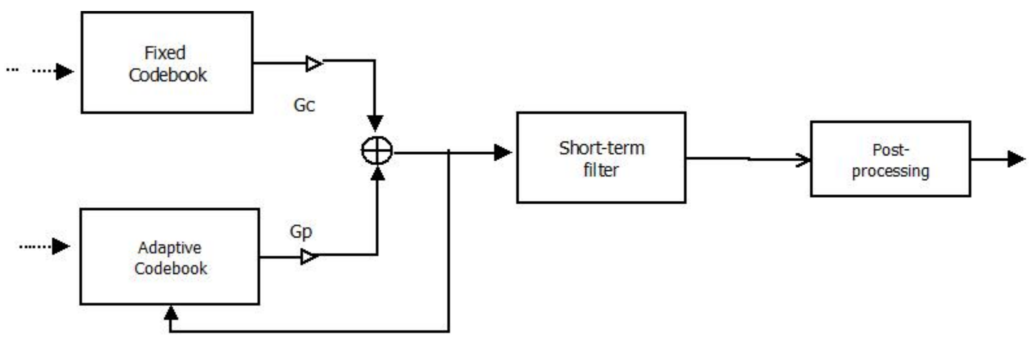


Figure 1.4: CS-ACELP Decoder: Decoding Principle [1]

The encoding and decoding principle is explained in this section in which ACELP is used as the basic principle. The idea of code comes from the fact that excitation codebook used in this vocoder contains the "code" to "excite" the " synthesis filter ". In this way voice coding and decoding known as vocoder is used to compress the data from 8000 samples to 80 bits per frame. This help in reducing the bandwidth required for storing the bits.

The thesis aims to develop the architecture for the Levinson durbin recursion and LP to LSP conversion, since the computational complexities of these modules are found to be more. The estimation of the parameters of the synthesis filters coefficients is done by LD algorithm which is an estimation of moving average model. Many speech vocoder consists of the linear analysis block in which the parameters are estimated rather than computed. **Moving Average Model** is a

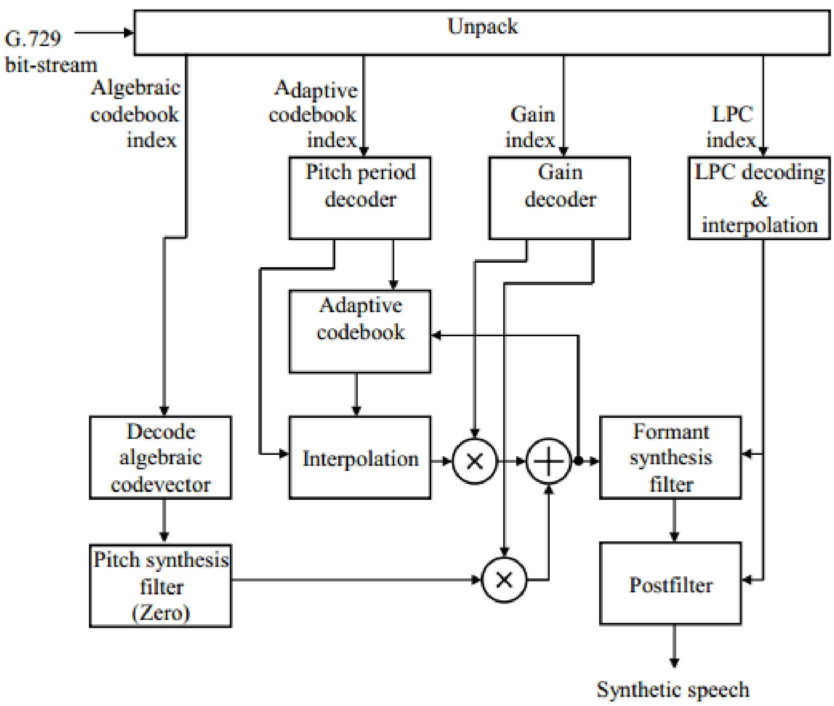


Figure 1.5: Decoding operation [12]

model in which the parameter are first estimated then transmitted to the next recursion and it follows till we get the desired order of prediction. In MA model there are many hurdles which include convergence problem of maximum likelihood estimation. Durbin algorithm is thus used to reduce this problem by estimating the parameters step-by-step rather than computing exhaustively.

1.3 Research Motivation

Vocoder is voice coding and decoding technique that are used for coding the human speech at various bit rates. They are widely used in the application of speech coding such as mobile telephony and voice over IP. Secure speech communication is desired in addition with low bit rate. Low bandwidth of speech signal can be achieved by coding the speech in almost 80 bits. Thus it reduces the bandwidth requirement.

Voice coding enables a telephonic company to transmit more signals at a time with single fibre; thus providing the multiplexing of the voices. For a mobile user, when the lower bit rate is provided, more number of voice signals can be accommodated in a single fibre. Voice coding is a lossy coder, since at the decoder side the reconstructed voice is different from the original voice. Thus the aim of voice coding is to determine a technique that will provide a reconstructed output which is close to the original one.

ITU recommendation VOCODER G.729 is a vocoder used for coding of the human speech. The aim of the coding speech is to protect the "intelligibility" of the speech. This software implementation or the hardware software co-design has found to be computationally costly. To make the coder computational effective with providing a trade-off between area and power complete hardware implementation is required.

The hardware implementation of speech signals are complex thus optimization techniques have to be provided to reduce the algorithmic delay. Various methods are provided in the literature that need to made efficient by proving the parallelism that will increase the speed of the coder. Linear prediction block is the block that is common in mostly available voice coding techniques.The LD algorithm in the LP analysis consumes more time in the software implementation. The algorithmic delay of this block has found to be more which need to be minimized.

1.4 Thesis Organization

The Thesis is organized into seven chapters. Each chapter's brief outline is given as follows:

Chapter 1: This chapter introduces the VOCODER G.729. It provides the description of encoder and decoder part of VOCODER G.729. The pros and cons of the speech coding is also discussed. This chapter also gives the research motivation that provides the reason for the thesis.

Chapter 2: This chapter describes the literature review on human speech production system,linear predictive coding,algorithmic approaches for linear predictive coding,VLSI implementation approaches, literature review on line spectral pairs.

Chapter 3:This chapter gives research gaps and research objectives and detailed description of research methodology .

Chapter 4: This chapter gives results analysis of the MATLAB implementation of VOCODER G.729 mainly focusing on Levinson Durbin algorithm and LP to LSP conversion blocks and codebooks of VOCODER .

Chapter 5:This chapter provides VLSI architecture for Levinson Durbin algorithm: Direct and Pipeline method. In addition, VLSI architecture of linear prediction coefficients to line spectral pairs is explained. This chapter also gives the simulation results and analysis. Area, power and delay computed for the blocks are described in this section.

Chapter 6: This chapter summarizes the key findings and main contributions of the Thesis done and the future scope.

Chapter 2

Literature Review

2.1 Human Speech Production Model

The CS-ACELP algorithm is based on the human speech production system [12].

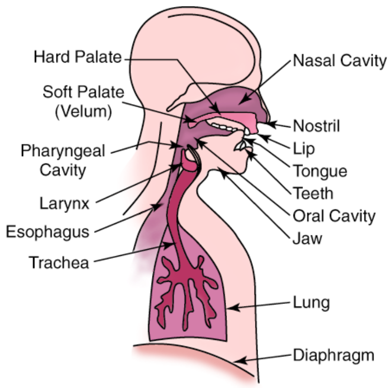


Figure 2.1: Human Speech Production System [12]

2.1.1 Discrete Time Model of Human Speech Production System

Two types of speeches are defined : voiced speech and unvoiced speech.Voiced speeches are generated through impulse generator and then passed through glottal filter which is a two pole filter.Unvoiced speeches are produced through random noise generator.This two speeches are passed through vocal tract model which is an all-pole filter.Finally the speech is passed through lip radiation model.

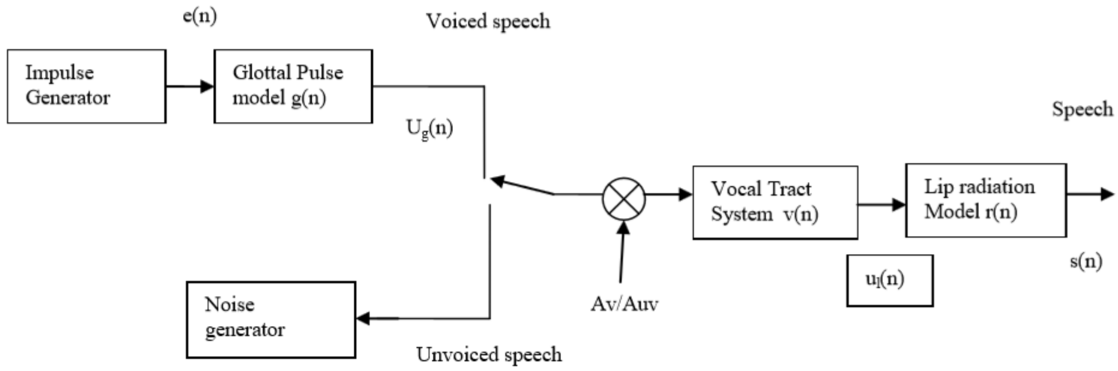


Figure 2.2: Discrete Time Model for Speech Production [12]

Speech can be divided into two classes, voiced and unvoiced.The difference between the two signals is the use of the vocal cords and vocal tract(mouth and lips).

For Voiced Speech:

$$\frac{S(z)}{E(z)} = A * G(z) * V(z) * R(z) \tag{2.1}$$

$$\frac{S(z)}{E(z)} = A\frac{1}{(1-z^-1)^2}\frac{1}{1+\sum_{k=1}^{P} az^-k}(1-z^-1) \tag{2.2}$$

$$\frac{S(z)}{E(z)} = \frac{A}{(1+\sum_{k=1}^{P+1} az^-k)} \tag{2.3}$$

For Unvoiced Speech

$$\frac{S(z)}{E(z)} = A * 1 * \frac{1}{1+\sum_{k=1}^{P+2L} az^-k}(1-z^-1) \tag{2.4}$$

2.1.2 Vocal Tract Model

For voiced speech,the "vocal tract" model can be realized by an "all-pole" model. The transfer-function for the vocal tract will be: **Overall Transfer Function For Vocal Tract Model**

$$V(z) = \frac{A}{(1+\sum_{k=1}^{P} az^-k)} \tag{2.5}$$

For unvoiced speech zeros are introduced in V(z) and for every zero it can be replaced by 2 poles.Vocal tract model is widely used in modern low bit rate speech coding algorithms,as well as speech synthesis and speech recognition systems.It is necessary to develop a technique which allows the coefficients of the model to be determined for a given frame of speech. The most commonly used technique is linear predictive coding (LPC).

2.2 Linear Predictive Coding

"Linear prediction" is an important part in almost all the latest modern day algorithms form "speech coding". The main principle of the linear prediction is when a speech signal is given it can be computed as the linear combination of its previous samples or past samples. Using mean-square prediction error; the weights that are used to compute the linear combination can be minimized within a single signal frame. The linear prediction coefficients or the resultant weights represent a particular frame [1].

It is a procedure that is used to estimate the auto regressive parameters of a given frame. In this way, LP is basically an identification technique in which the parameter of the module is found from observation. I addition the linear prediction helps in computation of the parameters of the AR model that is used to define the Power Spectral Density of a particular signal. This estimation is carried out by the line spectral pair compuation (LSP).

The advantage of computing the LPCs of a signal frame, it is possible that one can generate another signal that is approximately closer to the original one. LP is used for low bit rate transmission and storage. LPC is mostly referred as "inverse filtering" because it aims to determine the "all zero" filter that is inverse of the "vocal tract model". The input to the model will be the previous speech samples and the output is estimated current sample.

We need a technique to produce the coefficients of the model (α_k) such that they are equal to the coefficients of the speech production model (a_k).
There are some modified linear predictors that are used for linear prediction that are: "Warped Linear Prediction", "Two-sided Linear Prediction", "Frequency

Weighted Linear Prediction", "Discrete All-Pole Modelling", "Perceptual Linear Prediction", "Linear Prediction by sample grouping".

Linear Prediction Analysis block in VOCODER G.729

Linear Prediction block consists of windowing and autocorrelaion block,levinson durbin recursion block,LP to LSP block,quantization and interpolation block and finally LSP-LP block.

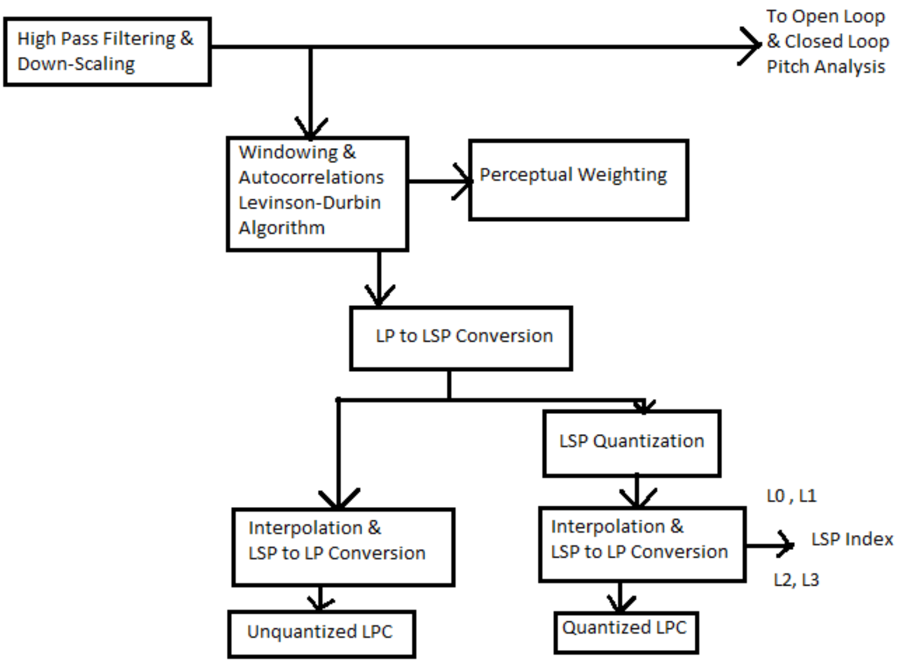


Figure 2.3: LP Analysis

2.2.1 The Auto-Correlation Method

The "autocorrelation-function" of a signal can be defined as

$R[k] = \sum_{m=k}^{+\infty} y[m]y[n-m] where k = 0, 1, 2....P$

The Matrix representation: $R(j) = \sum_{k=1}^{P} R(j-k)a_k$

$$
\begin{bmatrix} R(1) \\ R(2) \\ . \\ . \\ . \\ R(P) \end{bmatrix} = \begin{bmatrix} R(0) & R(1) & . & . & R(P-2) & R(P-1) \\ R(1) & R(0) & . & . & . & R(P-2) \\ . & . & & & . & . \\ . & . & & & . & . \\ . & . & & & . & . \\ R(P-1) & R(P-2) & . & . & . & R(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ . \\ . \\ . \\ a_p \end{bmatrix}
$$

where R(0)=Energy of the signal.

It is necessary to invert the auto-correlation matrix (R) in order to determine the LPC coefficients $a = R^{-1}r$.But this task can be quite cumbersome,given that it is a PxP matrix,with P typically between 10 to 14 in most applications.

Toeplitz Matrix: The matrix represented above is a square matrix which is known as Toeplitz matrix. A matrix is said to be toeplitz if the elements in the main diagonal are equal and the elements that are parallel to the main diagonal are are also equal. There are two methods defined in literature to solve this type of marices: "Levinson-Durbin Algorithm" and "Leroux-Gueguen Algorithm."

2.2.2 Levinson Durbin Algorithm

Levinson Durbin algorithm is a process in which the estimation of the parameters of the synthesis filters is done with an recursive method. The linear predictive computation is done which is the inverse of the synthesis filter. Input to LD algorithm is modified autocorrelation coefficients rr_k. Output are reflection coefficients(k),LPC coefficients(α_m) and short time energy (J). For the Mth order prediction of Levinson Durbin approach find the solution from the (M-1)th order predictor. This method is an "iterative-recurssive" process in which the solution to the first prediction order is done by first finding the solution to the "zero order predictor". This procedure is repeated until the order is reached that is desired.

Procedure of Levinson Durbin Algorithm:

1. **Predictor of order zero:**

$$R_{(0)} = J_{(0)}$$

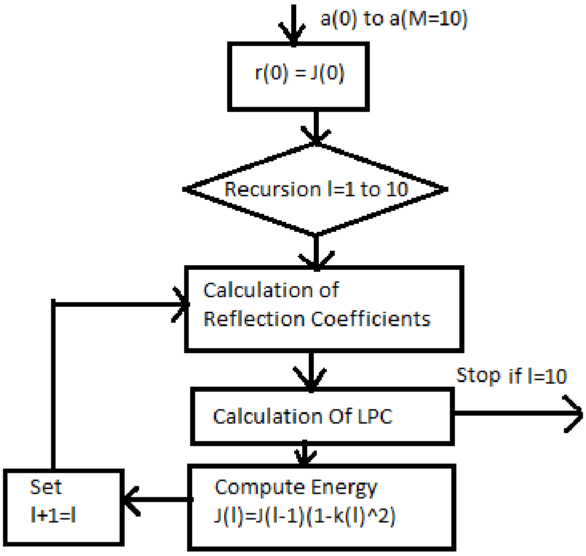


Figure 2.4: Flowchart of Levinson Durbin Algorithm

This equation provides a relationship between the minimum "mean-square prediction error" which is nothing but the autocorrelation coefficient at lag zero. This coefficient is found from the "zero order predictor". This value also defines the variance of the signal given. The "prediction error" is equal to the signal itself; because for a "zero order predictor" the prediction calculated is always zero.

$$\begin{bmatrix} J_{(0)} \\ \bigtriangledown_{(0)} \end{bmatrix} = \begin{bmatrix} R_{(0)} & R_{(1)} \\ R_{(1)} & R_{(0)} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

On solving $\bigtriangledown_{(0)} = R_{(1)}$

2. **Predictor of order one:**

$$\begin{bmatrix} J_{(1)} \\ 0 \end{bmatrix} = \begin{bmatrix} R_{(0)} & R_{(1)} \\ R_{(1)} & R_{(0)} \end{bmatrix} \begin{bmatrix} 1 \\ a_1^{(1)} \end{bmatrix}$$

$J_{(1)}$ describes the "minimum mean-squared prediction error" achievable using a "first order predictor". Here two unknowns- $a_1^{(1)}$ and $J_{(1)}$. Now considering a solution of the form

$$\begin{bmatrix} 1 \\ a_1^{(1)} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} - k_{(1)} \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

The "correlation matrix" is then multiplied on both sides:

$$\begin{bmatrix} J_{(1)} \\ a_1^{(1)} \end{bmatrix} = \begin{bmatrix} J_{(0)} \\ \bigtriangledown_{(0)} \end{bmatrix} - k_{(1)} \begin{bmatrix} \bigtriangledown_{(0)} \\ J_{(0)} \end{bmatrix}$$

On solving we get:

$$k_{(1)} = \bigtriangledown_{(0)}/J_{(0)} \text{ and}$$
$$J_{(1)} = J_{(0)} - k_{(1)}\bigtriangledown_{(0)} = J_{(0)}[1 - k_{(1)}^2]$$

where k is reflection coefficient which is an other way for representing LPC. Expanding to dimension three:

$$\begin{bmatrix} J_{(1)} \\ 0 \\ \bigtriangledown_{(1)} \end{bmatrix} = \begin{bmatrix} R_{(0)} & R_{(1)} & R_{(2)} \\ R_{(1)} & R_{(0)} & R_{(1)} \\ R_{(2)} & R_{(1)} & R_{(0)} \end{bmatrix} \begin{bmatrix} 1 \\ a_1^{(1)} \\ 0 \end{bmatrix}$$

On solving we get:

$$\bigtriangledown_{(1)} = R_{(2)} + a_1^{(1)} R_{(1)}$$

The above procedure is repeated until the prediction order desired is achieved. In this the LD algorithm provides a recursive method. A ten order prediction is mainly used in the "speech coding" algorithms where the LPCs are transmitted by quantizing it as a information on a frame. This order is chosen in order to obtain a quite well power spectral density of an "unvoiced frame". The envelope of the power spectral density can be minimized by LD approach described above.

2.2.3 Leroux-Guegeun Algorithm

"Leroux" and "Gueguen" proposed a method in which the "reflection coefficients" can be computed directly from the "autocorrelation" values without dealing with the LPCs directly. Dealing with the environment of fixed-point implementation does not provide the accurate results. This problem, can be eliminated by this method.

Leroux-Gueguen Versus Levinson-Durbin

1. The "Leroux-Gueguen" algorithm is most often used for fixed-point implementation since, the intermediate values are known.
2. The "Leroux-Gueguen" approach followed by "RC-to-LPC" has a disadvantage that it does not provide saving computationally when compared to "Levinson Durbin algorithm".

2.2.4 Line Spectral Pair and its conversion from LPCs

"LINE SPECTRAL PAIRS" have found to be widely used in the applications where robust representation of LPCs are required. The filter responses of quantization procedure is represented more suitably by the properties of the LSP and the LSFs (Line Spectral Frequency). LSFs are used in voice compression as it has got its quantization property, that are computed by the LPCS.

While finding the LSPs two polynomials are first made P(z) and Q(z) where one is symmetric and the other is antisymmetric. These polynomials have got several properties that are interesting and useful. If roots of these equations are monotonically increasing and if these two polynomials are interleaved, then the stability of the filter can be ensured. In addition, filter is more resonant at the corresponding frequency when the two roots are closer. Since effect quantization noise is less in LSPs and stability can be easily ensured thus LSPs are most commonly alternative way of representing the LPCs. The procedure for converting a given LPC to LSP is given in figure below.

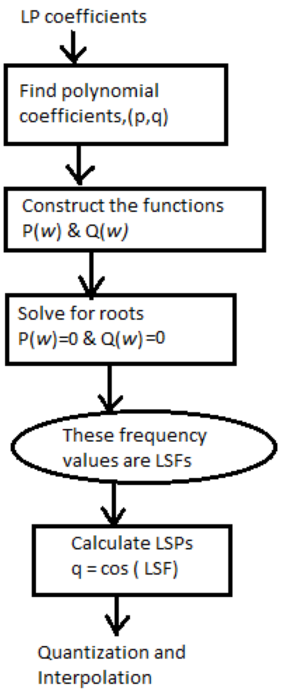


Figure 2.5: Flowchart for LPC-LSP Conversion

2.3 Algorithmic Approaches of Levinson Durbin Algorithm

[2] proposes an adaptive multi-stage Levinson-Durbin algorithm. The LD algorithm is broken into several stages adaptively, till the procedure becomes unstable. The residual signal of the previous stages are used in the estimation of parameters of the next stages. The two-step method introduced in this paper is found to be an improvement in the gain prediction than the conventional single stage prediction. It also shows that when linear predictor models are cascaded then more accurate model can be formulated for the input voice signal.

[3] proposes a technique in which LD algorithm is divided into stages of similar blocks in which either of the block can be used for processing. The number of multiplications and the number of additions required by split LD approach is nearly half than that is required by the conventional LD algoithm. Split method also helps in providing parallelism in the procedure.

[4] proposes a technique in which a two step levinson durbin algorithm is first

applied. Then the algortihm is extended to even-odd split levinson algorithm. This method further reduces the multiplications and additions as either even parameters are found first or the odd parameter.

2.4 VLSI Implementation Approaches

[5] provides the architecture for segmentation block,autocorrelation block,reflection logic block and the LPC coefficients block.The levinson durbin algorithm is divided into two logic blocks,one for reflection logic that provides the nominator and denominator logic and these are converted to linear predictive coefficients.

FPGA implementation for the LD algorithm based LPC for voice recognition is described. The described circuitry is synthesized using Xilinx VirtexE v2000ebg560 technology which is a 40nm technology and synthesis result uses the clock frequency used is 13.4Mhz. The resource utilization rate is given in figure.

Resource	Used	Avail	Utilization
IOs	178	404	44.06%
Global Buffers	1	4	25.00%
Function Generator	32508	38400	84.66%
CLB Slices	16254	19200	84.66%
Dffs or Latches	6538	40812	16.02%
Block RAMs	0	160	0.00%

Figure 2.6: The FPGA utilization rate [5]

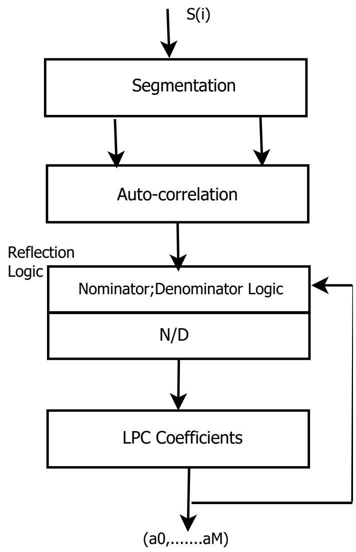


Figure 2.7: Block Diagram for LPC Feature Extraction [5]

[6] proposes a design for hardware implementation of LPC feature extraction system based on Levinson-Durbin algorithm, was proposed.Using registers be-

tween computational logic blocks,that exploits pipelining capability that results in increasing the clock rate and throughput of the system.

[7] proposes a hardware implementation of voice coding of CS-ACELP for compression of voice signals. The algorithmic delay of each hardware-implemented LP analysis and quantization was found to be 2,18,928 cycles. This implementation was done on Spartan 3 FPGA and a reference clock of 50 MHz.

2.5 Literature Review on Linear Prediction Coefficients to Line Spectral Pair Conversion

[8] proposes a direct conversion in the LSP domain,which ensures low computational costs. LSP domain perceptual weighting enables to weight the envelopes adaptively among the frequencies with a small amount of computation.

[9] describes the faster conversion of LPC to LSF method and its implementation.It takes only half the number of iterations for LPC-to-LSF conversation for 10th order compared to cubic based solution.It does not require any iteration after finding the root, as needed in complex free Ferrari based solution.

[10] describes method to improve the computation of LSPs, the enhanced Tschirnhaus transform (ETT) is proposed to accelerate the coordinated polynomial solution.A pipeline-recursive framework is implemented to save calculations. A 40-fold improvement in throughput is achieved.Reduces 1.16% of gate counts at the hardware synthesis level.

Chapter 3

Research Gaps, Research Objectives and Research Methodology

3.1 Research Gaps

The implementation of voice coding are divided into mainly three methods in literature: Software design, Software-hardware co design and hardware design. All these three methods had certain advantages and disadvantages.The software design though has advantage of high accuracy and low terminal noise since the implementation is carried out in floating point environment. However this software method shows large algorithmic delays. It requires sufficient time for the execution.

To overcome this software hardware co-design is done. In this only the modules consuming more time are implemented in hardware. However, this method has the disadvantage because both the software and hardware work in different environments. One in floating point and the other fixed point environment. Thus it affects the accuracy and distortion may be there in the output. This method however faster than software design still shows algorithmic delays.

The last method described in the literature is the hardware design. In this functionality of the modules are realized in the hardware domain in the fixed point environment. The methods described in the literature shows large area and more power consumption. Also since the implementation is done in fixed point the accuracy is less compared to software design.

3.2 Research Objectives

The research objective is to design an efficient architecture for LPC feature extraction for speech that is based on Levinson-Durbin algorithm and LP to LSP coeficients that will provide the trade off between area, power and latency. The primary objective is the functional verification that is to be done with the software designs such as in MATLAB. Corresponding identification of the modules consuming more time and developing an efficient architecture for it.

3.3 Research Methodology

In this section research methodology which will be adopted to carry-out the research work is discussed. Every step of research process is briefly shown in as follows: Description of each step of the proposed research methodology is discussed below:

1. Literature Review:
    - Make a list of selected Journals related to each blocks of VOCODER G.729.
    - Select the articles of interest.
    - Manage articles in tabulated form (S.No., Year, Journal, Abstract,Key Issues, Authors and Reference).
    - Short the list with relevance of topic.
2. Problem Identification or Research Gaps: On the basis of the highlighted issues in Step 1, research gaps are identified.
3. Formulation of Objectives: Objectives are formulated according to the research gaps identified in step 2. In the proposed methodology, designed Objectives are as follows.

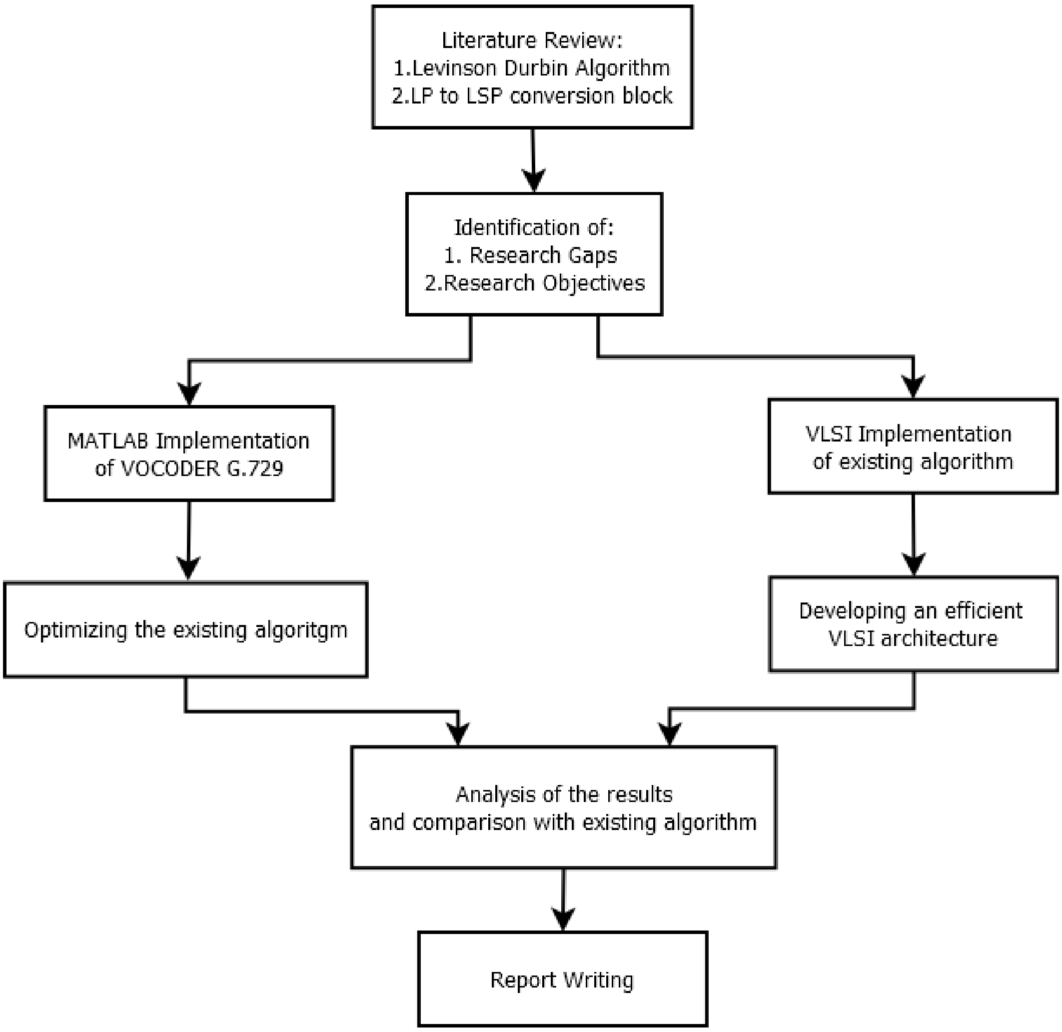


Figure 3.1: Research Methodology

- VLSI implementation of the algorithm is performed and an architecture to be proposed for the modified algorithm.
- Testing the modified or newly developed algorithms over the existing algorithms

4. Analysis of the results: Results are analyzed in terms of accuracy, efficiency and reliability. If the results are

5. Compare the obtained results with those of existing algorithms. Make some conclusions on the basis of comparison about accuracy, efficiency and reliability.

Chapter 4

Implementation Of Voice Coding in MATLAB

4.1 Linear Prediction Analysis

Input speech is a 8000 samples/sec and 16 bit PCM signal(Figure:4.2a).The processes that are taken are described as follows:

1. **Pre-processing:** Input Speech taken which is a PCM signal with 8000 samples per second is first pre-processed by down-sampling by factor then high pass filtering.The filter design is shown in Figure: 4.2b.
   The filter equation is given as:

   $$H(z) = \frac{0.46363718-0.92724705z^-1+0.46363718z^-2}{1-1.9059465z^-1+0.9114024z^-2}$$

   The filterd output is shown in Figure4.2c.

2. **Windowing and Autocorrelation Block:** The filtered output is segmented into 10ms frames and 5ms subframes.Total of 240 samples are taken into consideration for a window.Then auto-correlation is performed and 10 auto-correlation coefficients are obtained.A 60Hz bandwidth expansion is applied to these auto-correlation coefficients.The auto-correlation coefficients and modified auto-correlation coefficients are shown in Figure:4.2d.

3. **Levinson Durbin algorithm:** The input to LD algorithm is auto-correlation coefficients and outputs are the linear prediction coefficients that lie within

the unit circle (Figure:4.3a).By-product of this procedure are reflection coefficients that can be used for determining the vocal tract area.The error signal obtained (Figure4.3d) is minimum for voiced speech and maximum for unvoiced speech.

4. **LP-LSP conversion:** LP coefficients are then converted to LSP coefficients for quantization and interpolation.The Line spectral pairs that must lie on the circle which ensures the stability of the speech after quantization and interpolation.

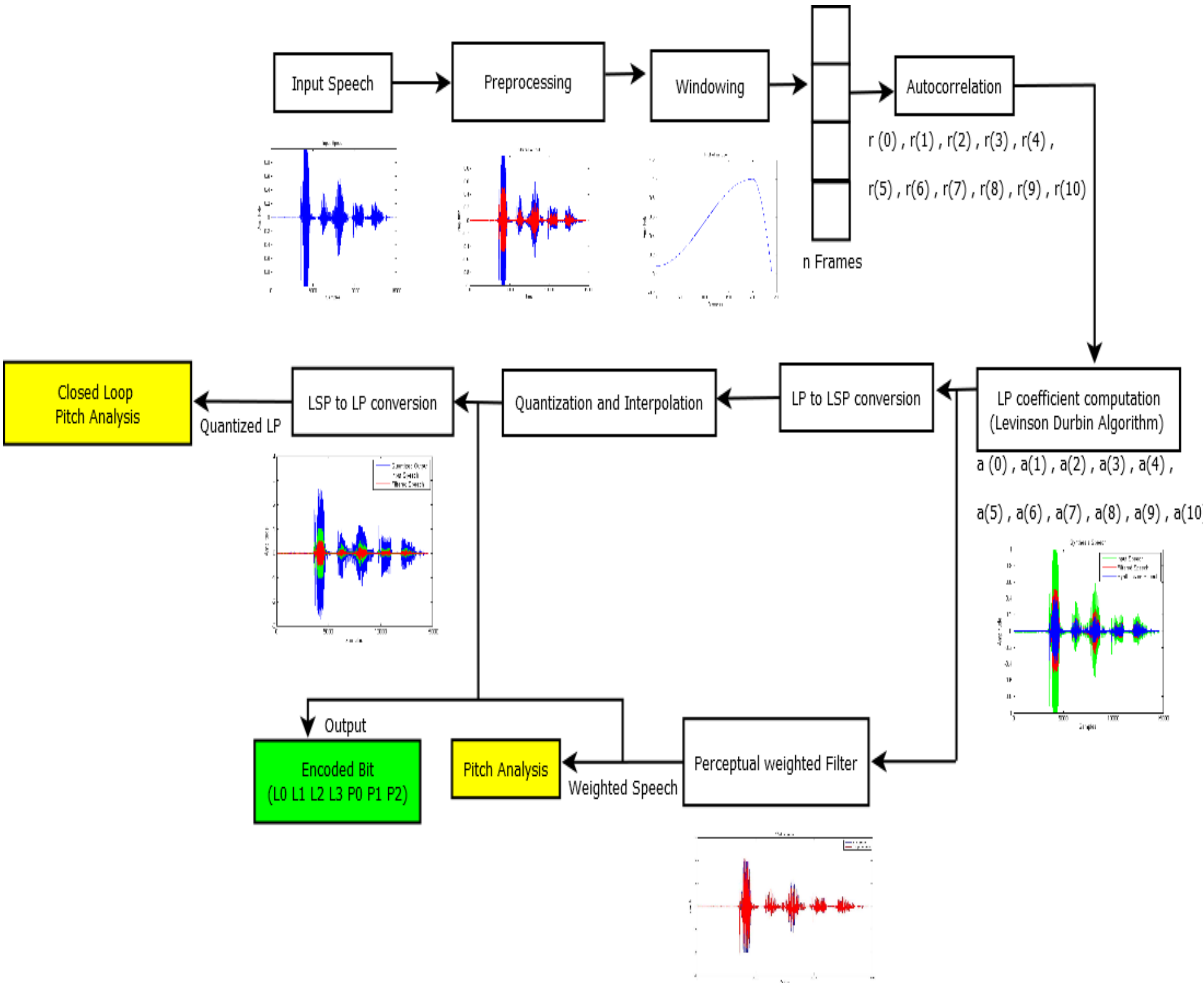
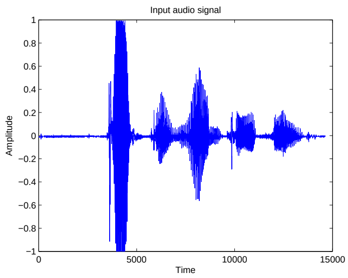
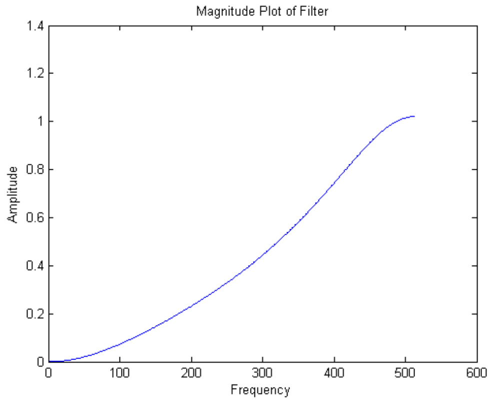


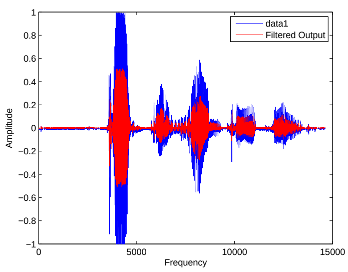
Figure 4.1: Process of LP analysis



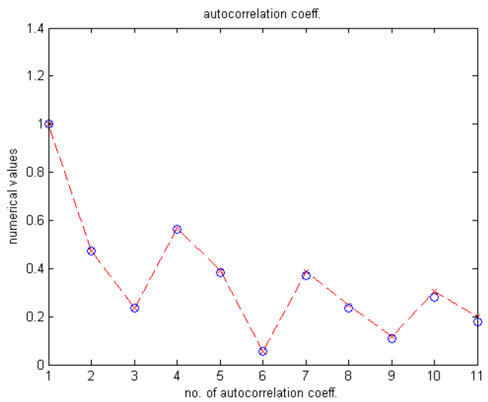
(a) Input Speech



(b) Filter

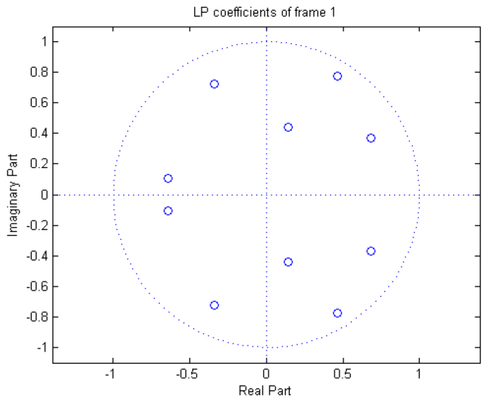


(c) Filtered Output

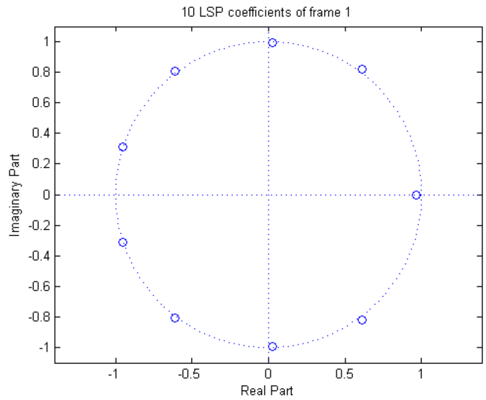


(d) Auto-correlation coefficients

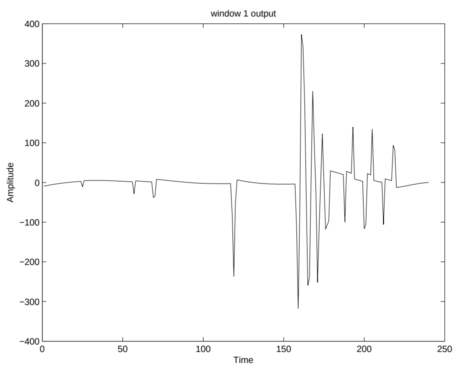
Figure 4.2: MATLAB Outputs



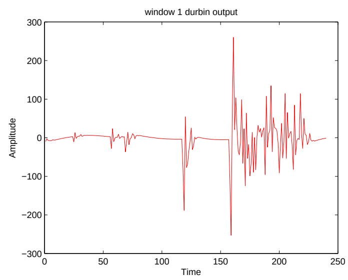
(a) LP coefficients



(b) LSP coefficients

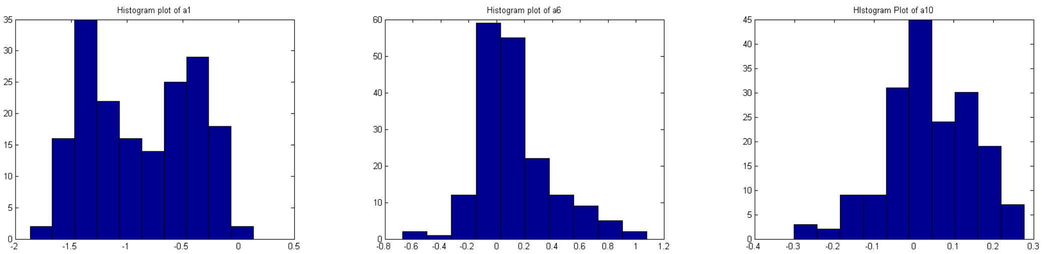


(c) Window 1



(d) Error signal

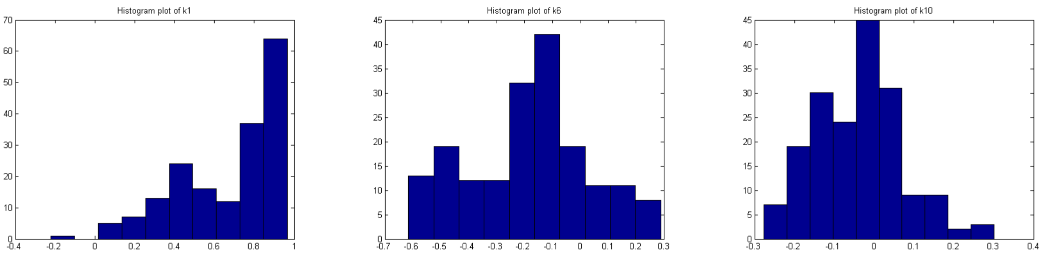
Figure 4.3: MATLAB Outputs



(a) (b) (c)

Figure 4.4: Some LPCs Histogram plots:(a)a_1 coefficient (b)a_6 coefficient (c) a_{10} coefficient

From the histogram plot we can observe that the lower-order coefficients have a dynamic range that is wider that that of higher order coefficients. The magnitude of the higher order coefficients also tend to be small, in addition they are gathered around the origin.



(a) (b) (c)

Figure 4.5: Reflection coefficient's Histogram plots:(a)k_1 coefficient (b)k_6 coefficient (c) k_{10} coefficient

Under a "fixed-point" environment the $|k_1|$ are bounded to be less than one. This bound is necessary in order to designing the algorithm efficiently.

4.2 CODEBOOKS : Adaptive Codebook and Fixed Codebook

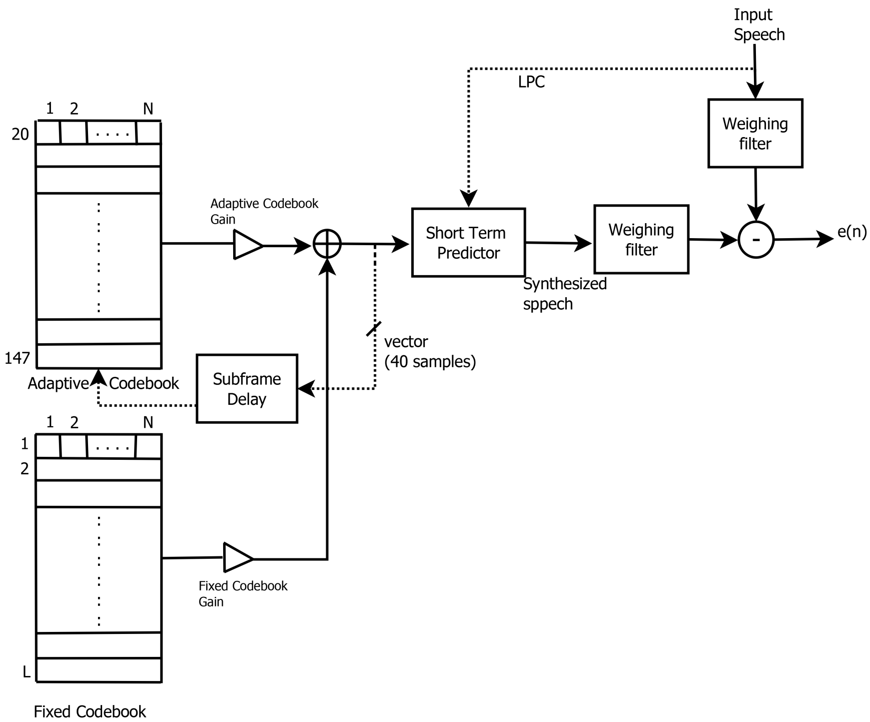


Figure 4.6: Block Diagram of Encoder illustrating codebook

Typically, the encoding is performed in the following order:

1. Linear Prediction Coefficients (LPC) are computed and quantized, usually as Line Spectral Pairs.

2. The adaptive (pitch) codebook is searched and its contribution is removed.

3. The fixed (innovation) codebook is searched.

The name of code-excited comes from the excitation codebook, containing the code to excite the synthesis filters.
What type of signal is contained in the excitation codebook? The codebook can be fixed or adaptive and can contain deterministic pulses or random noise. The length of each excitation code-vector is equal to that of the sub-frame (5ms). The adaptive codebook contribution provides periodicity (pitch) for the excitation, and the algebraic codebook contribution provides the remainder.

4.2.1 Adaptive Codebook - Structure and Search

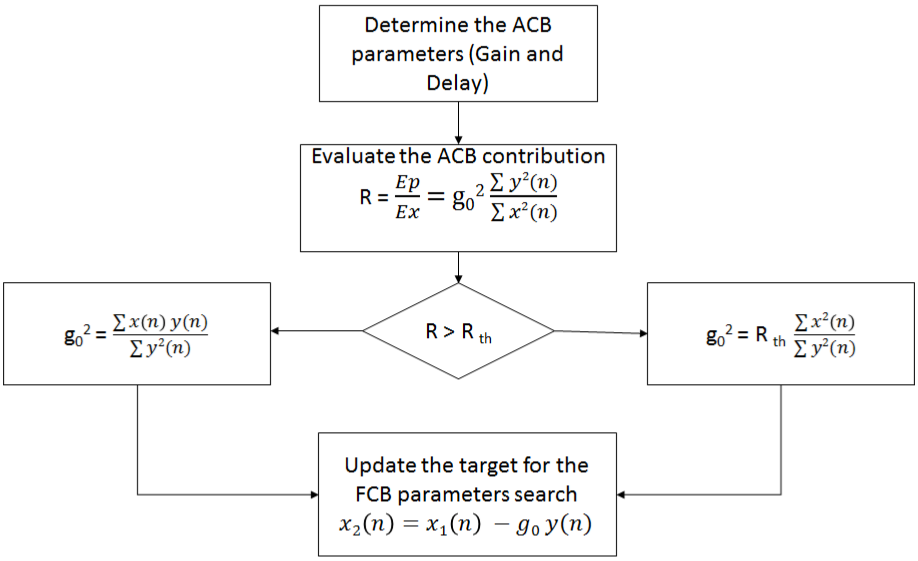


Figure 4.7: Adaptive Codebook Procedure

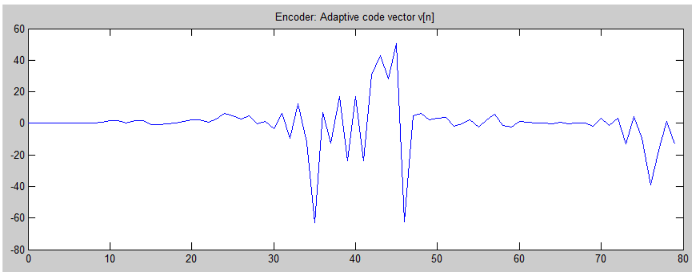


Figure 4.8: Adaptive Codevector

To search for the excitation parameters, first a target signal x1(n) is built by weighting the input speech and removing the zero-input response of the weighting-synthesis filter. The optimization is done sequentially. First, the optimal delay of the ACB is found then the optimal gain. To allow the decoder to recover faster after a frame erasure, the optimization is modified as follows:

First, the ACB contribution is evaluated by calculating the ratio between its energy and the energy of the target. If this is greater than the threshold, the gain of the ACB is modified to limit the contribution of the ACB.The result of this is that part of the long term contribution already modeled by the ACB will remain in the new target x2, and then it will be modeled as well by the ICB.This leads to a redundant representation of the pitch excitation.

4.2.2 Fixed Codebook - Structure and Search

- The G.729 coder has an algebraic structure for the fixed codebook that is based on the interleaved single-pulse permutation design (ISPP).
- There is no need to physically store the entire codebook resulting in significant memory saving.
- Input to fixed codebook :

  Target Signal x(n) : It is the output of adaptive codebook search which consists of 40 samples.

  Impulse Response h(n) : The impulse response of the weighted synthesis filter W(z)/A(z) is needed for the search of adaptive and fixed codebooks. It consists of 40 samples.

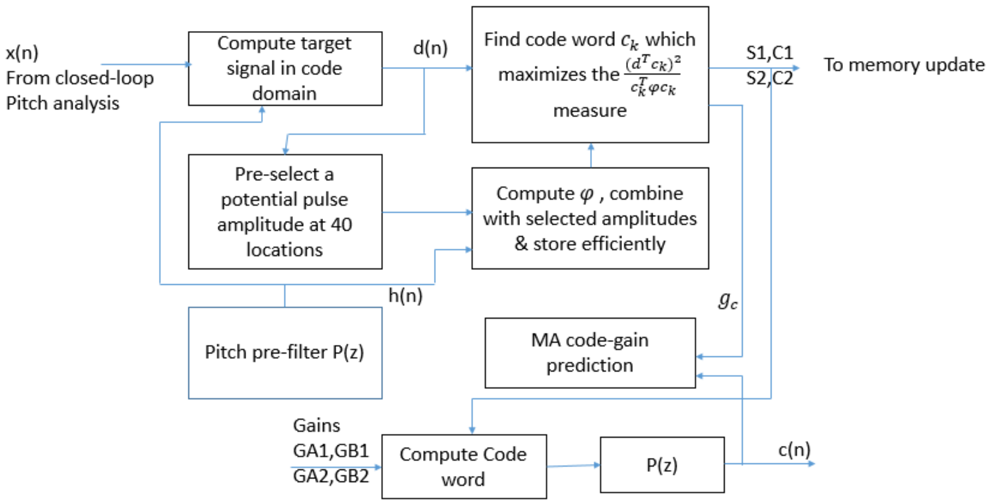


Figure 4.9: Algebraic Codebook Search(Fixed Codebook)

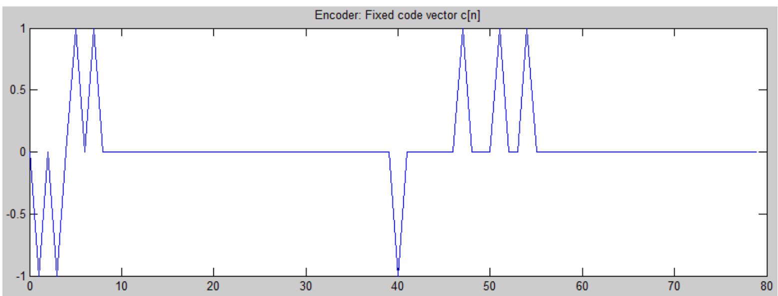


Figure 4.10: Fixed Codevector

Chapter 5

VLSI Architecture of Levinson Durbin Algorithm

5.1 Architecture of Levinson Durbin Algorithm

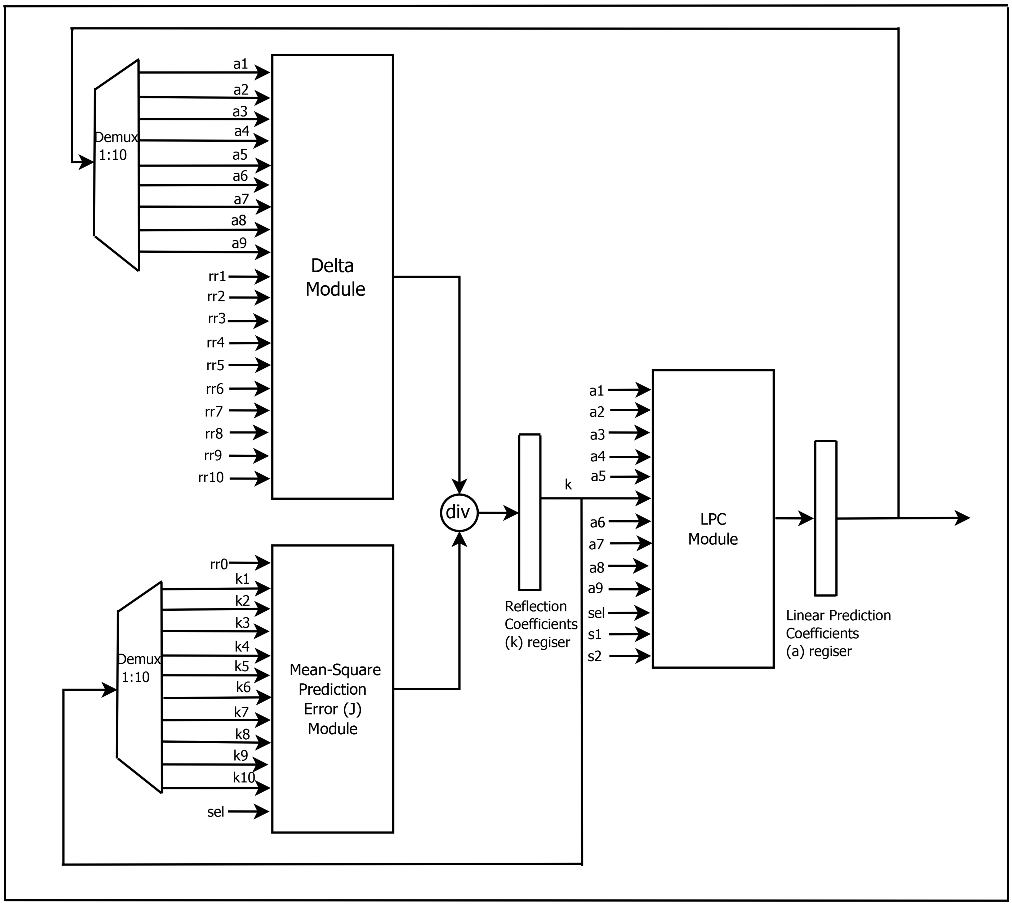


Figure 5.1: Architecture of Levinson-Durbin Algorithm

The architecture of the Levinson Durbin algorithm is as shown in the Figure 5.1. Algorithm is divided into the sub-modules for computing delta values, mean square values, and the LPC values. Each module is first implemented using the conventional method based on the algorithm. Then the corresponding area and power optimization is provided by proposing the pipeline method.

5.1.1 Generation of Delta Values

The numerator part for computation of the reflection coefficients are termed as delta value. For the input speech file the autocorrelation values are computed, these values are used for computing delta values. The LPC values that are calculated for previous recursion are also input to this module. The architecture is given for 10^{th} order linear prediction. Firstly a direct method is shown which uses large number of multiplies and adders, so a pipeline method is proposed for generation of delta values that uses minimum number of the adders and multiplies. The two methods are explained in the next subsections.

1. **DIRECT METHOD:**

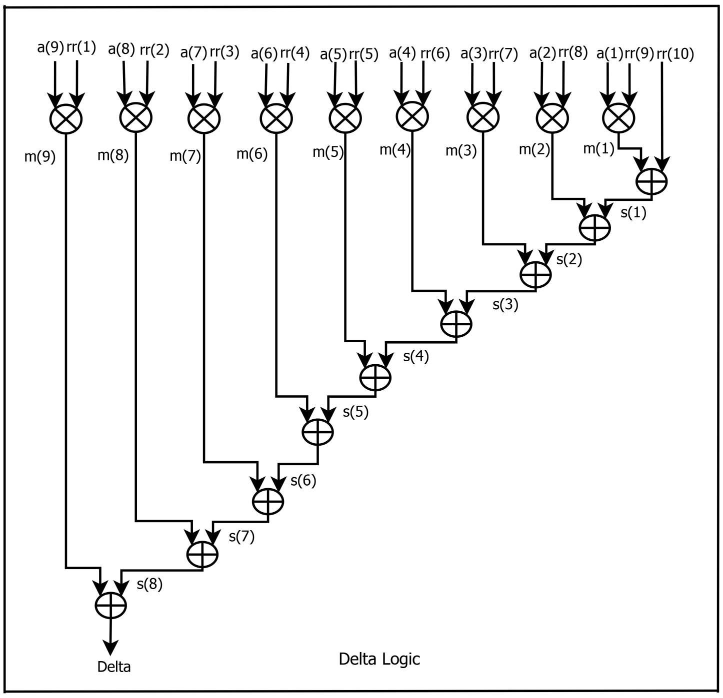


Figure 5.2: Delta Module

- The first step is calculation of reflection coefficients :
  $k = [R(l) + a_1^{(l-1)} R(l-1) + ... a_l^{(l-1)} R(1)]/J_{(l-1)}$

- $\bigtriangledown = R(l) + \sum_{i=1}^{l-1} a_i R(l-i)$
  $\bigtriangledown_0 = rr(1)$
  $\bigtriangledown_1 = rr(2) + a_1^1 rr(1)$
  $\bigtriangledown_2 = rr(3) + a_1^2 rr(2) + a_2^2 rr(1)$
  $\bigtriangledown_9 = rr(10) + a_1^9 rr(9) + a_2^9 rr(8) + a_3^9 rr(7) + a_4^9 rr(6) + a_5^9 rr(5) + a_6^9 rr(4) + a_7^9 rr(3) + a_8^9 rr(2) + a_9^9 rr(1)$

2. **PIPELINE METHOD:**

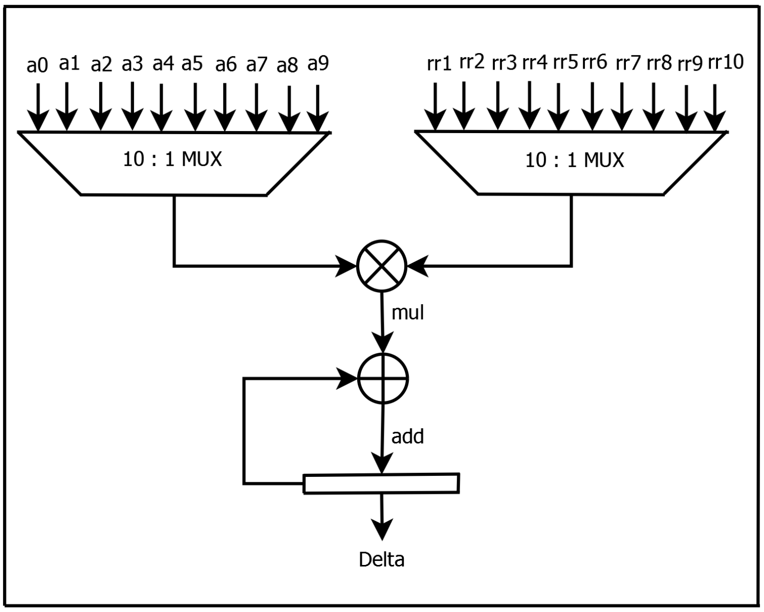


Figure 5.3: Pipeline Method of Delta Module

The advantage of pipeline method is it uses only one multiplier and one adder thus minimizing the area requirement. With proper selection of select lines of the multiplexers the autocorrelation coefficients and initial LPCs are selected. The output of the multiplexer is fed to the multiplier. The multiplier output is given as :

$$M_{(i)} = a_{(i)} * rr_{(l-i)} \textbf{ where l=10 (order of prediction)}$$

The multiplier output is added with the adder output giving a pipeline procedure. The output of adder is:

$$add_{(i)} = add_{(i-1)} + M_{(i)}$$

The output of the delta module is stored in the output register delta. The final output is given as:

$$Delta_{(l)} = rr_{(l)} + \sum_{i=1}^{l-1} a_{(i)} * rr_{(l-i)}$$

5.1.2 Mean Square Prediction Error Module

Mean square values are the denominator of the reflection coefficient computation denoted by J values. The mean square values are calculated for finding the best LPC value with minimum error and that is close to the original speech. Mean square values of the previous recursion and the reflection coefficient of the current recursion are used for calculating mean square values. Two methods are described first the direct mapping of the algorithm and next the pipeline method in which number of adders and multipliers used are reduced.

1. **DIRECT METHOD:**

   - The second step is calculation of mean square prediction error (J).
   - $J_{(0)} = rr(0)$
$k_1 = \bigtriangledown_0 / J(0)$
$J_{(1)} = J_{(0)}(1 - k_1^2)$
$J_{(10)} = J_{(9)}(1 - k_{10}^2)$
where $k_{10} = \bigtriangledown_9 / J_{(9)}$

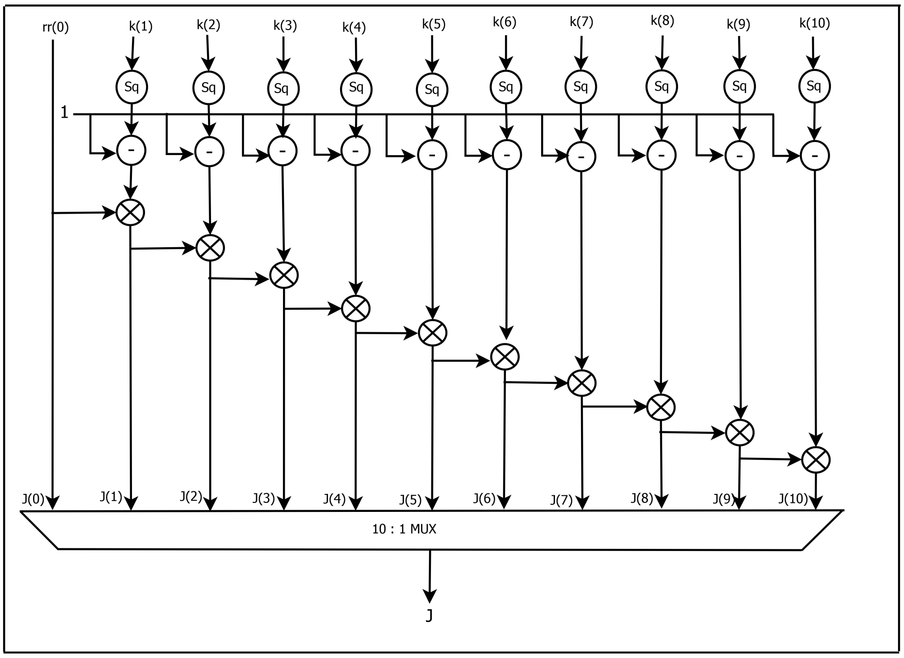


Figure 5.4: Mean Square Prediction Error Module

2. **PIPELINE METHOD:** The pipeline method for calculating mean square error values substantially decreases the area as number of multiplier and adder used are decreased greatly. In calculation of mean square error the reflection coefficients are fed to 10:1 multiplexer, with proper selection of multiplexer the output is given to a squarer and then subtracted by one. The output of subtracter is given as :

$$sub_{(i)} = 1 - k_{(i)}^2$$

Then mean square error values are calculated by pipeline method as:

$$J_{(i)} = J_{(i-1)}(1 - k_{(i)}^2)$$

The final value of the 10^{th} order prediction is given as :

$$J_{(10)} = J_{(0)}(1 - k_{(1)}^2)(1 - k_{(2)}^2)(1 - k_{(3)}^2)(1 - k_{(4)}^2)(1 - k_{(5)}^2)(1 - k_{(6)}^2)(1 - k_{(7)}^2)(1 - k_{(8)}^2)(1 - k_{(9)}^2)(1 - k_{(10)}^2)$$

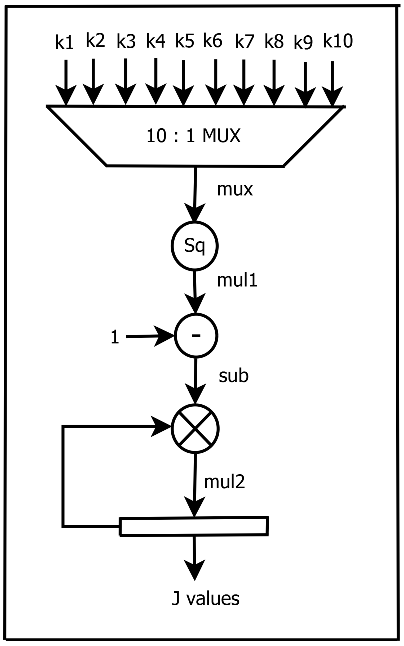


Figure 5.5: Mean Square Prediction Error Module: Pipeline Method

5.1.3 Generation of Linear Predictive Coefficients

The next step is to calculate linear predictive coefficients for the lth-order predictor. For every recursion the values calculated are used for next recursion till the desired order is reached. Here order is 10. The LPC value is calculated using the following equation:

$$a_l = -k_l, a_i^l = a_i^{l-1} - k_l a_{l-i} l - 1$$

For 10-order predictor the values are calculated as:

$$a_1^{10} = a_1^{10} - k_{10} a_9^{10},\ a_2^{10} = a_2^{10} - k_{10} a_8^{10},\ a_3^{10} = a_3^{10} - k_{10} a_7^{10},\ a_4^{10} = a_4^{10} - k_{10} a_6^{10},$$
$$a_5^{10} = a_5^{10} - k_{10} a_5^{10},\ a_6^{10} = a_6^{10} - k_{10} a_4^{10},\ a_7^{10} = a_7^{10} - k_{10} a_3^{10},\ a_8^{10} = a_8^{10} - k_{10} a_2^{10},$$
$$a_9^{10} = a_9^{10} - k_{10} a_1^{10}$$

These are the values that are used for forming the synthesis filter in the speech production model which is a short term prediction. In this way the speech feature extraction is done by determining the linear predictive coefficients.

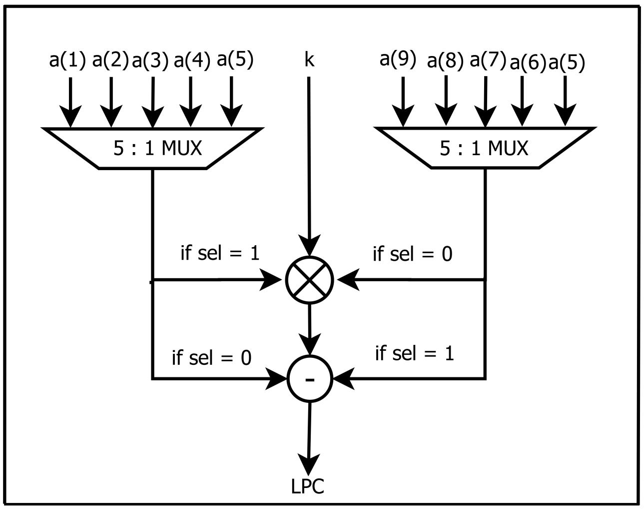


Figure 5.6: LPC Module

5.1.4 Levinson Durbin Algorithm for 10^{th} order Linear Prediction

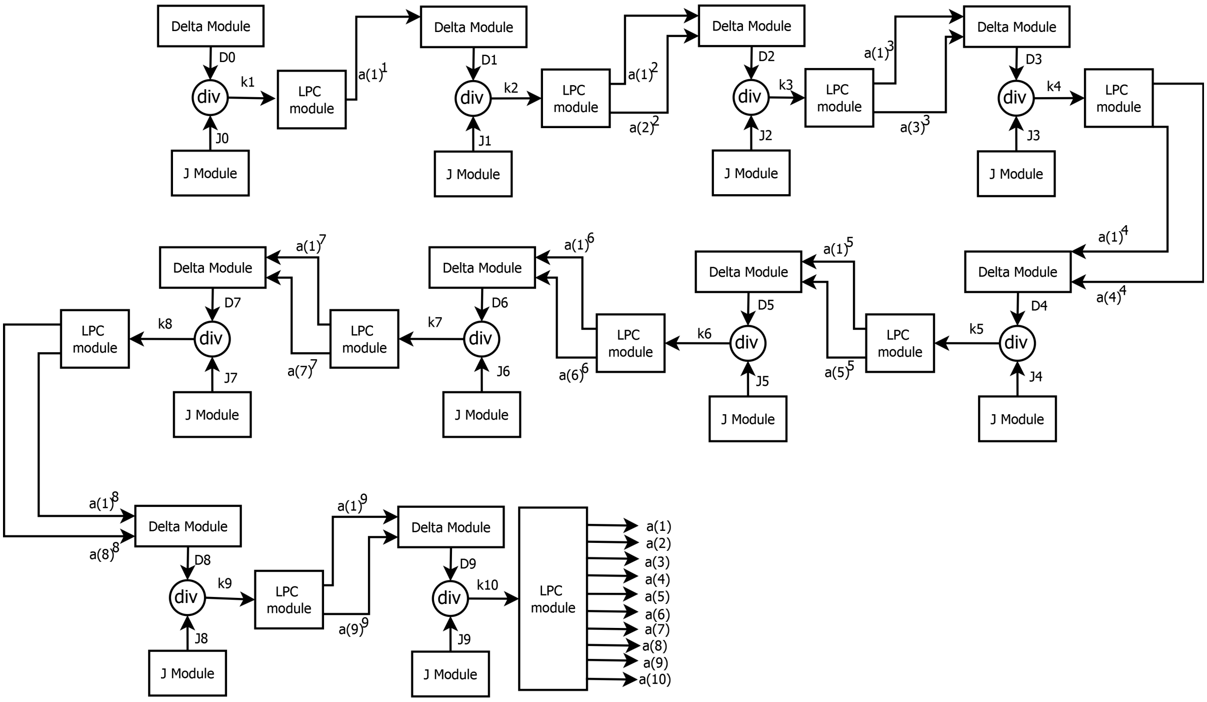


Figure 5.7: Levinson Durbin ALgorithm

5.2 VLSI Architecture for LP coefficients to LSP coefficients conversion

Line Spectral Frequency are used to represent linear prediction coefficients(LPC) for transmission over a channel.LSPs have several properties that make them superior to direct quantization of LPCs.For this reason,LSPs are very used in speech coding.

DEFINITION OF LINE SPECTRAL FREQUENCY: We are given the prediction error filter with system function :

$$A(z) = 1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} + ... + a_1 z^{-1}$$

5.2.1 Polynomial Generator

First phase in conversion of LP coefficients into LSP coefficients is the polynomial generator.The function of this block is to produce the coefficients for producing symmetric and antisymmetric polynomials.

$$\textit{Symmetric Polynomial: } P(z) = 1 + p_1 z^{-1} + p_2 z^{-2} + .. + p_M z^{-M} + z^{M+1}$$
$$\textit{Antisymmetric Polynomial: } Q(z) = 1 + q_1 z^{-1} + q_2 z^{-2} + .. + q_M z^{-M} + z^{M+1}$$

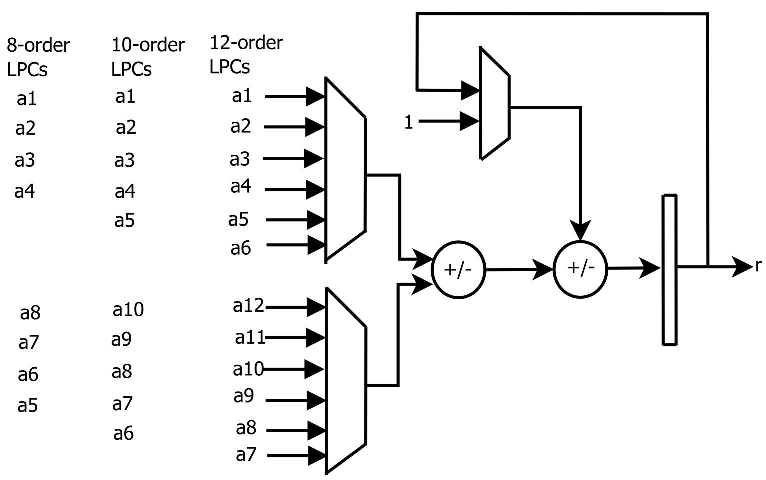


Figure 5.8: Architecture of symmetric/antisymmetric polynomial generator

5.2.2 General Form Polynomial

The second phase is the transformation of these polynomials into general form polynomials. General Form Polynomial :

$$f(x) = x^4 + \alpha_1 x^3 + \alpha_2 x^2 + \alpha_3 x + \alpha_4$$

where $\alpha_1 = r_1/2, \alpha_2 = (r_2 - 4)/4, \alpha_3 = (r_3 - 3r_1)/8, \alpha_4 = (r_4 - 2r_2 + 2)/16$

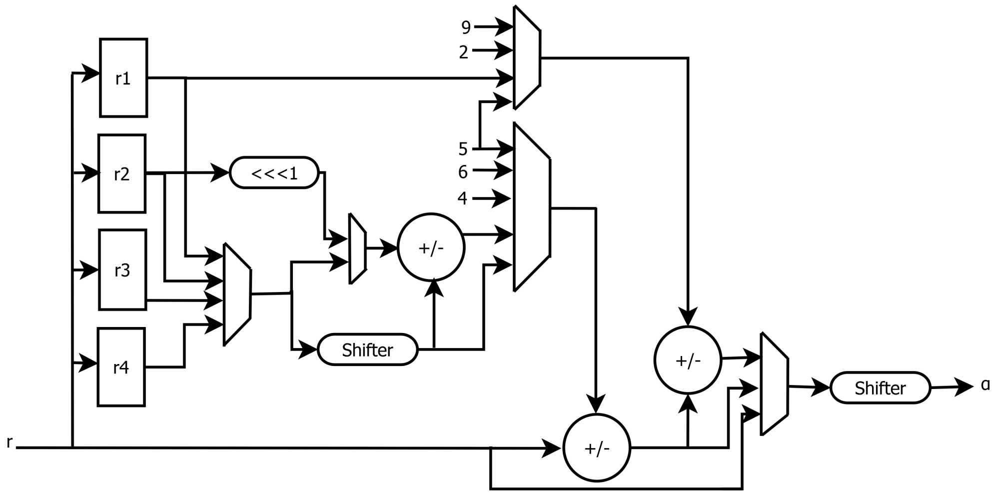


Figure 5.9: Architecture of general form polynomial generator

5.2.3 Iterative Root Finding Block

Iterative root finding block is designed based on the Birge-Vieta method which is a rapid root finding algorithm for polynomial solutions.It is a modified version of Newton-Raphson method.
Two polynomials that are obtained after general form polynomial are :

$$P(x) = 32x^5 + 16p_1x^4 + 8(p_2 - 5)x^4 + 4(p_3 - 4p_1)x^2 + 2(p_4 + 3p_2 + 5)x^4 + (p_5 - 2p_3 + 2p_1)$$
$$Q(x) = 32x^5 + 16q_1x^4 + 8(q_2 - 5)x^4 + 4(q_3 - 4q_1)x^2 + 2(q_4 + 3q_2 + 5)x^4 + (q_5 - 2q_3 + 2q_1)$$

where $x = cos(\omega)$

The architecture for LP to LSP conversion is implemented by referring [10]. This is done since line spectral pair is the better representation of the speech to analyse

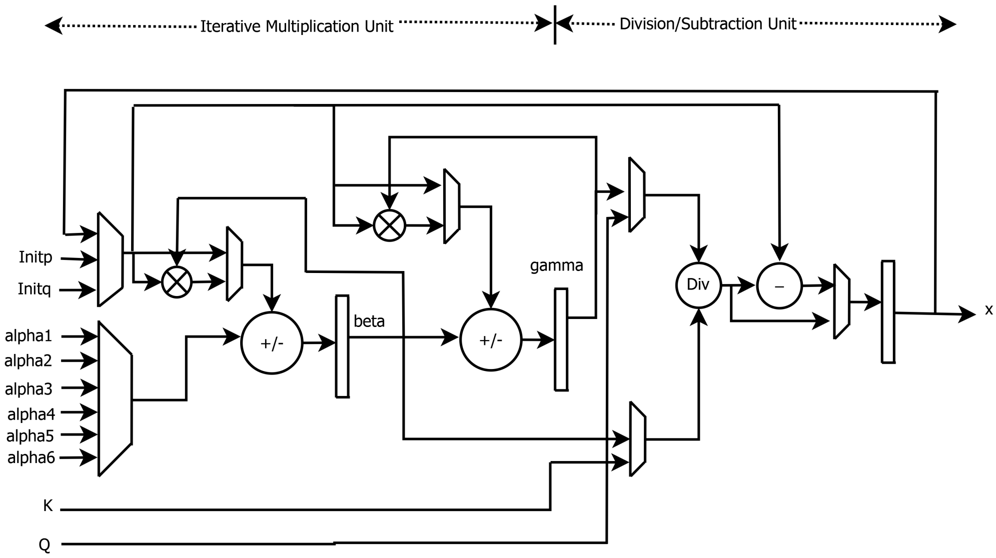


Figure 5.10: Iterative Root Finding Block

the input speech to do frequency analysis. The architecture is done to formulate of the symmetric and antisymmetric polynomials. The roots of the polynomial are then found in MATLAB by converting the 8 bit representation of the speech into decimal values since the root finding algorithm is complex.

5.3 Simulation results and analysis

This chapter presents the simulation analysis of the direct method LPC feature extraction of speech and the pipeline method. ASIC synthesis has been performed in Synopsys 90nm CMOS technology. The area report of the direct method LD algorithm and pipeline method are shown in tables 5.1 and 5.2 respectively which are calculated using Design Compiler. The power report of the direct method LD algorithm and pipeline method are shown in tables 5.3 and 5.4 respectively. Power reports are calculated using IC compiler.

Table 5.1: Area Report: Direct Method

Parameter	**Delta Module**	**J Module**	**LPC Module**	**Levinson Modlue**
No of Ports	160	108	93	201
No of Nets	314	369	135	305
No of cells	18	51	20	6
Cell Area (μm^2)	11464.70	22125.77	2033	39255.55

The area report shows that pipeline method of Levinson-Durbin algorithm is 3.22 times lesser than that of the direct method. Moreover, there is 38.9 times reduction

Table 5.2: Area Report: Pipeline Method

Parameter	Delta Module	J Module	LPC Module	Levinson Modlue
No of Ports	177	93	93	186
No of Nets	211	135	135	322
No of cells	12	27	20	22
Cell Area (μm^2)	2499.379	2661.407	2033.049	12175.257

Table 5.3: Power Report: Direct Method

Parameter	Delta Module	J Module	LPC Module	Levinson Modlue
Internal Power(mW)	4.117	6.163	0.507	4.7268
Switching Power(mW)	1.4077	1.898	0.15	1.6903
Leakage Power(mW)	0.168	0.3335	0.0293	0.548
Total Power (mW)	5.693	8.395	0.687	6.965

Table 5.4: Power Report: Direct Method

Parameter	Delta Module	J Module	Levinson Modlue
Internal Power(μW)	2.936	2.0829	2.876
Switching Power(μW)	0.798	0.5965	1.871
Total Power (μW)	43.259	41.485	179.02

in the power utilization of pipeline method than direct method. It shows that due to reduction in the hardware utilization the area and power are also reduced. IO utilization is shown in table 5.5.

Table 5.5: IO Utilization

S.No.	Module Name	IO Buffers	IO Utilization
1	**Delta Module**	108 out of 320	33%
2	**Mean Square Error Module**	100 out of 320	34%
3	**LPC Module**	93 out of 320	29%
4	**Levinson Durbin Algorithm**	185 out of 320	57%

Table 5.6: Area Report: LSP Module

Parameter	PolyGen	GenForm	IRFB	MCU
No of Ports	115	54	98	145
No of Nets	147	128	200	292
No of cells	5	10	13	19
Cell Area (μm^2)	2100.8	3414.683	5547.96	7086.98

Table 5.7: Power Report: LSP Module

Parameter	PolyGen	GenForm	IRFB	MCU
Internal Power(μW)	3.992	4.387	2.84	11.338
Switching Power(μW)	1.341	1.2182	0.989	3.5837
Leakage Power (μW)	29.35	48.059	79.53	100.79
Total Power (μW)	39.68	54.075	82.81	115.717

Computational Delay taken by VLSI implemented Levinson Durbin algorithm per frame is **77.215 nsec**. Computational Delay taken by MATLAB implemented Levinson Durbin algorithm per frame is **164.8 usec**. Area and power reports of the LP to LSP conversion are shown in tables 5.6 and 5.7. Multiplier free circuits are made for LSP modules as referring [10]. Thus the circuit utilizes less area. The above all analysis are done for an input speech signal which is a PCM signal. The input decimal values are first converted to binary representation then given to subsequent blocks. Since the analysis are done in fixed point environment, so there may be variation in the accuracy. Still approximate feature extraction of speech can be achieved.

Chapter 6

Conclusion and Future Scope

To understand the functionality of the VOCODER G.729, MATLAB implementation is done and corresponding modules are identified that takes more algorithmic delay. It is found that the LD algorithm and LP to LSP conversion takes more cycles. So an architecture is developed for these modules. The VLSI architecture of Levinson Durbin algorithm is divided into sub modules that makes the computation easy. This work is concerned with designing a VLSI architecture for conversion of linear predictive coefficients to line spectral pair and levinson durbin algorithm. Since line spectral pairs is the better representation of LP coefficients lsp representation is widely used. It is shown that due to reduction in hardware logic there is reduction in the area utilization and power consumption. The internal power and switching are of concern when it will be used in the VOCODER implementation since it consumes more power.

The design can be further minimized in area and power by using efficient multipliers and adders such as Booth Multiplier, Carry Save adders etc. The design is implemented in the fixed point environment thus there can be variation in the accuracy. The accuracy can be achieved by floating point implementation of design by the use of cordic architectures and systolic arrays.

KEY REFERENCES

[1] G. Draft Recommendation, "729–coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear prediction (cs-acelp)," *Telecommunication Standardization Sector, ITU-T*, 1995.

[2] R. Yu, X. Lin, and C. Ko, "A multi-stage levinson-durbin algorithm," in *Signals, Systems and Computers, 2002. Conference Record of the Thirty-Sixth Asilomar Conference on*, vol. 1, pp. 218–221, IEEE, 2002.

[3] P. Delsarte and Y. V. Genin, "The split levinson algorithm," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 34, no. 3, pp. 470–478, 1986.

[4] A. Melman, "A two-step even–odd split levinson algorithm for toeplitz systems," *Linear Algebra and its Applications*, vol. 338, no. 1, pp. 219–237, 2001.

[5] J. Xu *et al.*, "Migrate levinson-durbin based linear predictive coding algorithm into fpgas," in *2005 12th IEEE International Conference on Electronics, Circuits and Systems*, pp. 1–4, 2005.

[6] B. Fazlali and M. Eshghi, "A pipeline design for implementation of lpc feature extraction system based on levinson-durbin algorithm," in *Electrical Engineering (ICEE), 2011 19th Iranian Conference on*, pp. 1–5, IEEE, 2011.

[7] M. L. George, "Linear prediction analysis and quantization for the conjugate-structure algebraic-code-excited linear-prediction speech compression algorithm," 2012.

[8] R. Sugiura, Y. Kamamoto, N. Harada, H. Kameoka, and T. Moriya, "Direct linear conversion of lsp parameters for perceptual control in speech and audio

coding," in *Signal Processing Conference (EUSIPCO), 2014 Proceedings of the 22nd European*, pp. 56–60, IEEE, 2014.

[9] J. V. Domadiya, A. P. Kandagal, and N. Ramesh, "A complex ferrari lpc to lsf implementation," in *Contemporary Computing and Informatics (IC3I), 2014 International Conference on*, pp. 458–462, IEEE, 2014.

[10] C.-H. Chang, B.-W. Chen, S.-H. Chen, J.-F. Wang, and Y.-H. Chiu, "Low-complexity hardware design for fast solving lsps with coordinated polynomial solution," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 23, no. 2, pp. 230–243, 2015.

[11] T. Bäckström *et al.*, *Linear predictive modelling of speech: constraints and line spectrum pair decomposition.* Helsinki University of Technology, 2004.

[12] W. C. Chu, *Speech coding algorithms: foundation and evolution of standardized coders.* John Wiley & Sons, 2004.

[13] M. Atri, F. Sayadi, W. Elhamzi, and R. Tourki, "Efficient hardware/software implementation of lpc algorithm in speech coding applications," 2012.

[14] A. Kataoka, T. Moriya, and S. Hayashi, "An 8-kb/s conjugate structure celp (cs-celp) speech coder," *Speech and Audio Processing, IEEE Transactions on*, vol. 4, no. 6, pp. 401–411, 1996.

[15] N. Khan, M. I. Khattak, M. Khan, F. Khan, L. U. Khan, S. A. Salam, and D.-W. Gu, "Implementation of linear prediction techniques in state estimation," in *Applied Sciences and Technology (IBCAST), 2013 10th International Bhurban Conference on*, pp. 77–83, IEEE, 2013.

[16] F. Khan, N. Khan, L. U. Khan, M. Naeem Khan, and B. Pirzada, "On the optimal frame size of linear prediction techniques," in *Circuits, Power and Computing Technologies (ICCPCT), 2013 International Conference on*, pp. 331–336, IEEE, 2013.

[17] L.-M. Lee and H.-C. Wang, "Extended levinson-durbin algorithm for the analysis of noisy autoregressive process," *IEEE Signal Processing Letters*, vol. 3, no. 1, 1996.

[18] P. Vaidyanathan, "The theory of linear prediction," *Synthesis lectures on signal processing*, vol. 2, no. 1, pp. 1–184, 2007.

[19] H. Wakita, "Direct estimation of the vocal tract shape by inverse filtering of acoustic speech waveforms," *Audio and Electroacoustics, IEEE Transactions on*, vol. 21, no. 5, pp. 417–427, 1973.