Design and Analysis of Short Word Length DSP Systems for Mobile Communication

A Thesis Submitted in Fulfillment of the requirements for
the Degree of Doctor of Philosophy

Tayab Din Memon
PgD, B.ENG

School of Electrical and Computer Engineering
College of Science, Engineering and Health
RMIT University
June 2012

Declaration

I certify that except where due acknowledgement has been made, the work is that of the author alone; that work has not been submitted previously, in whole or in part, to qualify for any other academic award; the content of the thesis is the result of work which has been carried out since the official commencement date of the approved research program; and any editorial work, paid or unpaid, carried out by a third party is acknowledged.

Signed: ……..

Tayab Din Memon

Date:

Summary

In last decade, Short Word Length (SWL, often single-bit) processing has proved promising technique in the development of DSP applications with low complexity and high performance. Recently, many general purpose DSP applications such as Least Mean Squares-Like single-bit adaptive filter algorithms have been developed using this SWL technique and have been shown to achieve similar performance as multi-bit systems. The reported benefits of the SWL techniques include their intrinsic simplicity of operation, low power consumption and efficient hardware implementation.

A key function in SWL systems is sigma delta modulation (ΣΔM) that operates at an over sampling ratio (OSR), in contrast to the Nyquist rate sampling typically used in conventional multi-bit systems. Using large over sampling ratios is one way to improve the noise performance of a system, although potentially at the cost of overall throughput.

To date, the analysis of SWL (or single-bit) DSP systems has tended to be performed using high-level tools such as MATLAB, with little work reported relating to their hardware implementation, particularly in Field Programmable Gate Arrays (FPGAs). Two primary areas of interest exist here. The first is the comparative behaviour of SWL and multi-bit systems exhibiting at equal spectral performance in terms of their relative area, power and throughput. Secondly, it remains to be determined how chip area-performance varies with varying OSR and bit-width of the hardware SWL system.

This thesis explores the hardware implementation of single-bit systems in FPGA using the design and implementation in VHDL of a single-bit ternary FIR-like filter as an illustrative example. The impact of varying OSR and bit-width of the SWL filter has been determined, and a comparison undertaken between the area-performance-power characteristics of the SWL FIR filter compared to its equivalent multi-bit filter. Further, an analysis of single-bit adaptive channel equalization in MATLAB has been performed, which is intended to support the design and development of efficient algorithm for single-bit channel equalization.

As the performance of FIR filters is chiefly determined by the throughput of their multiply-accumulate (MAC) stages, an efficient organization for the design and implementation of this block has been proposed and its area-performance characteristics analysed using commercial FPGA devices in Quartus-II® and ModelSim®. It has been found that SWL filters can achieve clock frequencies in the range of 400 MHz suitable to process, for example, a 6 MHz video signal at an OSR of 64. The proposed adder organization has been used as a baseline for further investigation into the comparison between single-bit FIR-like filters and their conventional multi-bit counterparts.

The proposed ternary filter structure has been merged with IIR re-modulator component and the design and implementation of the overall single-bit FIR filter has been explored in hardware in order to compare its power-area-performance characteristics with approximately equivalent multi-bit FIR filters. Both filters types were designed and simulated in pipelined and non-pipelined mode. In this set of simulations, varying OSR (32 – 256) was used to identify the area-performance-power analysis of two techniques. The simulation results show that single-bit FIR-like filter consistently outperforms the multi-bit technique in terms of its area, performance and power except at the highest filter orders analysed in this work. It was also found that increasing OSR increases SNR at the cost of higher chip-area.

The stability of the single-bit FIR-like filter mainly depends upon IIR remodulator due to its recursive nature. Thus, we have investigated the stability IIR remodulator and propose a new model using linear analysis and root locus approach that takes into account the widely accepted second order sigma-delta modulator state variable upper bounds. Using proposed model we have found new feedback parameters limits that is a key parameter in single-bit IIR remodulator stability analysis.

In the second stage of thesis, three encoding techniques called canonical signed digit (CSD), 2's complement, and Redundant Binary Signed Digit (RBSD) were designed and investigated on the basis of area-performance in FPGA at varying OSR. Simulation results show that CSD encoding technique does not offer any significant improvement as compared to 2's complement as in multi-bit domain. Whereas, RBSD occupies double the chip area than other two techniques and has poor performance.

Finally, aspects of single-bit adaptive channel equalization, which is a key element in all the communication systems, have been analyzed. A new mathematical model has been derived with all inputs, coefficients and outputs in single-bit domain. The model was simulated using narrowband signals in MATLAB and investigated on the basis of symbol error rate (SER), signal-to-noise ratio (SNR) and minimum mean squared error (MMSE). The results indicate that single-bit adaptive channel equalization is achievable with narrowband signals but that the harsh quantization noise has great impact in the convergence.

Table of Contents

Declaration ii
Summary iii
Table of Contents v
List of Tables viii
List of Figures ix
Acknowledgments xi
Dedication xii
1. Introduction 1
1.1 Introduction 1
1.2 Research Questions and Thesis Objectives 3
1.2.1 Research Questions 3
1.2.2 Aims and objectives 3
1.3 Novel Contributions 4
1.1 Thesis Organization 6

2.Conventional and Single-bit FIR Filtering Techniques 8
2.1 Introduction 8
2.2 Sigma-Delta Modulation 14
2.2.1 Sigma-Delta Modulator Block Diagram 15
2.2.2 Trade-offs between ΣΔM OSR and Modulator Order 17
2.2.3 Quantizer Behaviour in ΣΔM 18
2.2.4 Linear and Non-Linear Models of ΣΔM 19
2.2.5 Sigma-Delta Modulator Z-domain Analysis 20
2.3 Fast and Efficient FIR Filter Design Techniques 27
2.3.1 Fast FIR Filters 29
2.3.2 Sigma-delta modulation based fast filters 29
2.4 Single-bit Filtering Techniques 35
2.4.1 Single-bit techniques 36
2.5 VLSI Analysis of ΣΔM based bit-stream circuits 42
2.5.1 FIR Filter Design Techniques in FPGAs 45
2.6 Role of FPGAs in mobile communication 47
2.7 Summary 49

3. Power-Area-Performance Characteristics of FPGA-based ΣΔ FIR Filters.....51

3.1 Introduction.....51

3.2 FIR Filter Design Techniques.....53

3.3 Single-Bit Ternary FIR-like filter.....56

3.3.1 Ternary FIR Filter (TFF).....57

3.3.2 Generation of Ternary FIR filter in MATLAB.....59

3.3.3 IIR Re-modulator.....61

3.4 FIR Filter Design in VHDL.....62

3.4.1 Single-bit Ternary FIR-like Filter Hardware Implementation.....62

3.4.2 Ternary multiplier and adder modules.....64

3.4.3 Multi-bit FIR filter design.....66

3.4.4 Spectral Performance Comparison.....67

3.5 Simulation Results and Discussion.....69

3.5.1 Filter Area-Performance Analysis.....69

3.5.2 Filter Power Analysis.....73

3.6 Stability Analysis of ΣΔM Based Single-bit IIR Filter.....79

3.7 Stability of Single-bit Ternary FIR-like filter.....82

3.8 Proposed Design of SBTFF.....83

3.9 Stability Analysis by Root Locus Technique.....86

3.10 Simulation Results and Discussion.....87

3.11 Summary.....90

4. FPGA Analysis of Sigma-Delta Modulated Ternary FIR Filter with Alternative Encoding Techniques.....93

4.1 Introduction.....93

4.2 The Ternary FIR Filter (TFF).....94

4.3 Noise Shaping in Sigma-Delta Modulators.....95

4.3.1 Ternary Filter Design in MATLAB.....98

4.4 Ternary FIR Filter Design in FPGA.....100

4.4.1 Two's-complement.....102

4.4.2 Redundant Binary Signed Digit (RBSD) Representation.....103

4.4.3 Canonical Signed Digit (CSD) Representation.....105

4.5 Simulation Results and Discussion.....108

4.6 Summary.....111

5. Single-bit Ternary Adaptive Channel Equalization for Narrowband Signals 114

5.1 Introduction.....114

5.2 System Design.....120

5.3 Single-bit ternary LMS-like Adaptive Channel Equalization Algorithm 123

5.3.1 Wiener Theory and LMS algorithm.....124

5.3.2 SBTLMS algorithm derivation.....127

5.4 Simulation and discussion 133

5.4.1 Symbol Error Rate (SER) at Varying input Training Samples 134

5.4.2 Signal-to-Noise Ratio (SNR).....135

5.4.3 Minimum Mean Squared Error (MMSE) 137

5.5 Summary 138

6. Conclusion and Future Directions.....140

6.1 Introduction.....140

6.2 Future Directions 145

List of Tables

Table 3.1 Signal-to-Noise Ratio Comparison of Single-bit and Multi-bit FIR Filter 67
Table 3.2 Area-Performance comparison of single-bit FIR vs. multi-bit filter: non- pipelined Mode. 71
Table 3.3 Area-Performance Comparison of Single-bit FIR vs. multi-bit Filter: pipelined Mode 71
Table 3.4 Clock Frequency for Ternary and Multi-bit Filters pipelined and non-pipelined modes. 75
Table 3.5 Dynamic Power Dissipation: F_{MAX} Process. 78
Table 3.6 Dynamic Power Dissipation: F_{8K} Process 79
Table 5.1 : IIR loop stability analysis using root locus with varying quantizer gain (γ) and feedback loop gain (α) parameters with and without proposed design 89
Table 5.1 Improvement in the SNR_o recorded with varying input SNR_i 138

List of Figures

Figure 2.1 General structure of FIR filter 9
Figure 2.2 Block Diagram of an IIR direct form II filter..... 10
Figure 2.3 General diagram of the sigma-delta modulator 16
Figure 2.4 Linear Model of Quantizer..... 19
Figure 2.5 First Order Sigma-Delta Modulator Topology 22
Figure 2.6 First Order Sigma-Delta Modulator Topology with Loop Filter Specified 25
Figure 2.7 NTF at $1 - 3^{rd}$ orders of the sigma-delta modulator..... 26
Figure 2.8 Block diagram of the error feedback ΣΔM for requantization 31
Figure 2.9 Block diagram of the FIR filter with ΣΔM modulated filter coefficients 32
Figure 2.10 Block diagram of the decoder used in FIR filter with ΣΔM modulated filter coefficients and with ΣΔM modulated input signal 32
Figure 2.11 Block diagram of the FIR filter with ΣΔM modulated input signal 33
Figure 2.12 Block diagram of the single-bit FIR filter 37
Figure 2.13 Block diagram of the first order single-bit IIR filter..... 38
Figure 2.14 General bock diagram of the single-bit ternary FIR filter..... 40
Figure 2.15 Single-bit narrowband bandpass FIR filter..... 40
Figure 2.16 First order digital sigma-delta modulator[50] 44
Figure 3.1 General Block Diagram of Single-bit FIR filter structure (adapted from [7])..... 54
Figure 3.2 Block diagram of Ternary FIR filter (adapted from [11])..... 54
Figure 3.3 Second Order ΣΔM architecture 55
Figure 3.4 Target Impulse Response by Remez Exchange Algorithm..... 60

Figure 3.5 Block Diagram of SBTFF in Hardware 63
Figure 3.6 Two Level Fragment of the Adder Tree Structure. 65
Figure 3.7 Frequency Response of the Target Filter at various coefficients bit-widths (=12, 16 and 18)..... 65
Figure 3.8 Proposed single-bit ternary filter with a gain factor inside the loop.... 81
Figure 3.9 ΣΔM quantizer input $g_2(k)$ at sinusoidal excitation 84
Figure 3.10 Linear Model of the 2nd order ΣΔM..... 88
Figure 3.11 Root Locus Plots with and without gain factor (ϕ) 90
Figure 4.1 Block diagram of Ternary FIR filter (adapted from [11]). 95
Figure 4.2 Linear Model of Quantizer..... 97
Figure 4.3 Target Impulse Response of FIR filter 99
Figure 4.4 Ternary Filter Impulse Response at OSR = 32, 64,128 and 256..... 99
Figure 4.5 Frequency Response of a Ternary FIR Filter 100
Figure 4.6 TFF hardware architecture 101
Figure 4.7 RBSD addition 104
Figure 4.8 Flow chart of the single-bit ternary CSD Multiplier 107
Figure 5.1. Equalizers types, structures, and algorithms [100]..... 118
Figure 5.2. Adaptive linear FIR equalizer with LMS algorithms [100]..... 119
Figure 5.3. General block diagram of an adaptive equalizer 122
Figure 5.4. Block Diagram of Single-bit ternary Adaptive Channel Equalization..... 122
Figure 5.5. Second order sigma delta modulator 127
Figure 5.6. General block diagram of single-bit block LMS-like filter [5] 129
Figure 5.7. The proposed SBLMS adaptive algorithm structure 133
Figure 5.8. SER at varying input training samples..... 135
Figure 5.9. SER recorded at varying input SNR(dB) 137
Figure 5.10. MMSE averaged over 1 to 30 trials 139

Acknowledgments

Firstly, I would like to thank Allah who created an opportunity for me to work as PhD candidate at RMIT, Melbourne. Without His support this was an impossible to move forward even a single step. Secondly, I would like to thank my whole family for their continued love and support whilst completing my studies. More important is my wife and three kids whose passions and support throughout this period allowed me to keep on the track and finish this work in the time.

Special thanks to Dr. Paul Beckett, my supervisor and more than a mentor, again without your help and drive I would never have finished this thesis; I value your input, way of working, and arguments. Thanks to Dr. Zahir Hussain who gave a direction to work in this area and supported to kick off in the right direction. Also thanks to Dr. Amin Z Sadik, he has helped a lot throughout this project especially in understanding of single-bit adaptive theory.

I would like to thank my sponsor Mehran University of Engineering and Technology (MUET), Jamshoro and Higher Education Commission (HEC) of Pakistan, who provided funds for PhD studies at RMIT University Melbourne.

Finally I'd like also to thank RMIT University and staff for continual help and support especially in all administrative activities.

Tayab Memon

Dedication

This dissertation is dedicated to my wife (Fatima) and my daughters Ureba, Bareera, and Arfa.

Publications

Below is the list of publications that have resulted directly from the work undertaken by the author for this PhD thesis.

Journal Publications

1. Tayab D Memon, Paul Beckett, Amin Z Sadik, "Power-Area-Performance Characteristics of FPGA-based Sigma-Delta FIR Filters", *Journal of Signal Processing Systems Springer (JSPS)*, No. 11265, ISSN: 1939-8018, DOI: 10.1007/s11265-012-0664-8.
2. Tayab D Memon, Paul Beckett, Amin Z Sadik, "Efficient Implementation of Ternary SDM Filters using State-of-the-Art FPGA", *Mehran University Research Journal of Engineering & Technology*, Volume 30, No. 2, APRIL, 2011, ISSN 0254-7821.
3. Paul Beckett, Tayab Memon, "Reconfigurable Blocks Based on Balanced Ternary", *Journal of Signal Processing Systems Springer (JSPS)*, No. 11265, ISSN: 1939-8018, DOI: 10.1007/s11265-010-0559-5 (First published online).
4. Tayab Memon, Paul Beckett, "The Impact of Alternative Encoding Techniques on the FPGA Implementation of Sigma-Delta Modulated Ternary FIR Filter", *Institute of Engineers Australia (IEAUST) Electrical and Electronics Journal, E11-061 (**Accepted For Publication Feb 2012**)*.

Refereed Conference Publications

1. Tayab D Memon, Abdullah Al-Hassani, Paul Beckett, "Single-bit Ternary FIR Filter in FPGA Using Canonical Signed Digit", accepted for publication at *2nd International Multi-Topic Conference (IMTIC'12)*, Mehran University Jamshoro, on 28 – 30th March, 2012.

2. Tayab D Memon, Paul Beckett, "Ternary Sigma-Delta FIR Filters", *17th Asia and South Pacific Design Automation Conference (ASP-DAC),* Sydney 30th Jan – 2nd Feb 2012.
3. Tayab D Memon, Paul Beckett, Amin Z. Sadik, Peter O'Shea, "Single-bit Adaptive Channel Equalization for Narrowband Signals", *22nd IEEE Region 10 conference TENCON* (TENCON'12), Bali, Indonesia, Nov, 2011.
4. Tayab D Memon, Paul Beckett, Amin Z. Sadik, "Performance-Area Tradeoffs of Ternary and Conventional FIR filter in FPGA", *5th IEEE International conference on MEMS, NANO and Smart Systems (ICMENS)*, Dubai, UAE, 28-30th December, 2009.
5. Tayab D Memon, Paul Beckett, Amin Z. Sadik, "Performance-Area Tradeoffs in the Design of a Short Word Length FIR Filter", *5th IEEE International conference on MEMS, NANO and Smart Systems (ICMENS)*, Dubai, UAE, 28-30th December, 2009.
6. Tayab D Memon, Paul Beckett, Zahir M Husain, "Analysis and Design of a Ternary FIR Filter Using Sigma Delta Modulation", *13th IEEE International Multitopic Conference (INMIC'09)*, Islamabad, Pakistan, 2009.
7. Tayab D Memon, Paul Beckett, Zahir M. Hussain, "Design and Implementation of Ternary FIR filter using Sigma Delta Modulation", *International Symposium on Computing, Communication and Control (ISCCC'09)*, Singapore, October 9-11, 2009.

Presentations

1. Tayab D Memon, Paul Beckett, Amin Z. Sadik, "Single-bit and Conventional FIR Filter Comparison in State-of-Art FPGA", *Annual RMIT University HDR Conference, Melbourne,* October 2010.

Chapter – 1

Introduction

1.1 Introduction

Although rapid advances in Very Large Scale Integration (VLSI) have made it possible to implement fast and efficient DSP functions in hardware, there is a continuing pressure towards smaller area with high performance at low power consumption in portable devices. As a result, there has been much research into finding optimal hardware implementations that fulfil these competing requirements [1-4]. For example, the characteristics of Finite Impulse Response (FIR) digital filters, which are widely used in signal processing applications, depend directly on the complexity of the essential multiplication steps that, in turn, increase linearly with the order of the filter. Regardless of the many optimizations that have been proposed, a large number of multiplication stages still translates into large area, delay and power consumption.

Sigma delta modulation (ΣΔM) based systems have the potential to mitigate the overhead of large multiplications and reduce the complexity of modern DSP systems. Sigma-delta modulators, which have already been widely adopted for A/D or D /A conversion, have recently been utilized for the development of DSP applications. For example, a LMS-Like single-bit adaptive filter has been developed to address the issue of noise cancellation in the real time mobile applications [5].

Despite these advances, there are many issues to be resolved particularly the application of Short Word Length (SWL) systems to VLSI implementations and how these contrast to their equivalent multi-bit system. These issues are addressed in this thesis as a way of promoting the adoption of ΣΔM based SWL systems in both communications related and general purpose DSP systems.

The term SWL is generally used to represent a system whose input, intermediate signals and final output can be in short word format i.e., 1 – 3 bits. Often SWL systems are known by the terms (that are even used throughout this thesis are) *binary* (or single-bit or bit-stream), and *ternary*. In the case of a FIR-like filter (see section 3.3) and adaptive channel equalization (see section 5.3) the term *single-bit ternary* has been used to indicate that the filter coefficients are in ternary format while its input is in binary (or single-bit i.e., +1, -1) format.

Following is given the detailed research questions, prospect objectives and novel contribution in the domain of SWL DSP systems.

1.2 Research Questions and Thesis Objectives

This thesis set out to answer the questions outlined below. To this end, the work has focussed primarily on the hardware characteristics of SWL FIR filters, especially in FPGAs. In addition, single-bit adaptive channel equalization and stability analysis of single-bit ternary FIR-like filter has been addressed in MATLAB.

1.2.1 Research Questions

- How can efficient, fast single bit and ternary filters are organized based on Sigma Delta Modulation to be used for Lowpass, Bandpass and other applications in mobile communication?

- How do single-bit, ternary and multi-bit FIR filters exhibiting equivalent spectral performance compare in terms of their power-area-performance characteristics? In particular, what is the impact of increasing the OSR or bit-width in hardware?

- How can the best possible stability criterion of single-bit ternary FIR filter be achieved?

- Is it possible to utilize LMS adaptive techniques for the design and development of SWL LMS-like adaptive channel equalization? Can we achieve adaptive equalization using coefficients in a ternary format?

1.2.2 Aims and Objectives

The primary aims of this work have been to:

- Design an efficient algorithm for the development of ternary adder circuit in VHDL that can be adopted for the synthesis of ternary FIR filter and investigating its area-performance characteristics

- Analyze the power-area-performance characteristics of single-bit ternary FIR-like filter in FPGA at varying OSRs in order to compare it with its corresponding multi-bit FIR filter

- Design and propose a more reliable stability model for IIR remodulator that takes into account all the stability factors of ΣΔM and that can be applied to the single-bit ternary FIR-like filter

- Investigate the area-performance characteristics of single-bit Ternary FIR filter using the alternative techniques of 2's complement, Canonical Signed Digit (CSD) and RBSD.

- Design and investigate a new single-bit ternary adaptive channel equalization organization using block LMS algorithm

1.3 Novel Contributions

- In single-bit FIR filters the requirement for a high OSR rate tends to cause the multiply/accumulate stage to become bulky. As a result, an efficient adder circuit design has been proposed and analyzed for these structures;

- A novel method of finding the power-area-performance characteristic of single-bit and multi-bit FIR filter in hardware at equivalent spectral performance has been determined and used to compare the two approaches with varying OSR;

- A new model has been proposed for the stability analysis of single-bit IIR filter using linear analysis and a root locus approach. The model takes into account typical stability upper bounds and achieves better stability results with extended feedback parameters limits;

- An illustrative single-bit Ternary FIR filter has been designed, implemented and simulated on a commercial FPGA range and their area and performance behavior analyzed using 2's complement, canonical signed digit (CSD) and Redundant binary signed (RBSD) representations for data and coefficients;

- The non-trivial task of single-bit adaptive channel equalization has been addressed and a novel model has been proposed and simulated in MATLAB using narrowband signals. The results have shown significant achievement in terms of signal-to-noise ratio (SNR), symbol error rate (SER), and minimum mean squared error (MMSE).

1.1 Thesis Organization

This thesis is organized as follows. Chapter 2 provides a survey on sigma delta modulation based short word length signal processing techniques, highlighting previous work done in this area. Contemporary multi-bit applications, especially FIR filter design and implementation in FPGAs have been extensively addressed. The architecture level design of single-bit applications is briefly covered.

In chapter 3, we present the comparison of single-bit and multi-bit FIR filters in FPGAs on the basis of their power, area and performance characteristics. The comparative filters were coded in VHDL using pipelined and non-pipelined modes and simulations carried out with binary data streams and ternary coefficients. It was found that the single-bit FIR filter offers superior area performance tradeoffs except at very high filter order. Further, the stability of single-bit IIR filter organizations was investigated and a new design proposed. This takes into account the stability upper bounds and enhances the control over sigma-delta modulator and quantizer input so that it becomes easier to control the overall stability of the system. With the proposed design, the upper limits of the feedback parameter increases from 0.16 to 1.5.

In chapter 4, three alternative encoding techniques; Two'2 complement, canonical signed digit (CSD) and Redundant Binary Signed Digit (RBSD) were investigated for the representation of the coefficients and data. Simulations were carried out using small commercial available FPGAs from Altera. The area-performance characteristics of the ternary FIR filter were evaluated and maximum operating frequency (F_{MAX}) was

computed for each simulation. Simulation results show that, in contrast to the case with conventional filters, digit encoding techniques such as CSD do not offer significant advantages in the single-bit domain. RBSD has been show to consume twice the chip area and returns no performance advantage.

In chapter 5, a novel approach to single-bit adaptive channel equalization is proposed. All of the inputs are maintained in single-bit format including channel transfer function. MATLAB simulation results shows that equalization filter can achieve significant signal-to-noise ratio (SNR), very small symbol error rate (SER), and minimum mean squared error (MMSE) with narrowband signals. Further work on this topic may leads towards better performance with more features and accuracy.

In chapter 6 we conclude and point to future work.

Chapter – 2

Conventional and Single-bit FIR Filtering Techniques

2.1 Introduction

In general terms, two classes of digital filters are available: Finite Impulse Response (FIR) and Infinite Impulse Response (IIR). The choice of these filters can be categorized on the basis of speed, chip area, hardware complexity, spectral filtering and linear phase requirements. Both filters have advantages and disadvantages. FIR filters offer linear phase and simple hardware implementation but require a higher filter order to meet a specific application requirements compared to the IIR filter. By contrast, IIR filters exhibit stability problems due to their recursive nature that increases the overall filter gain and exaggerates quantization errors.

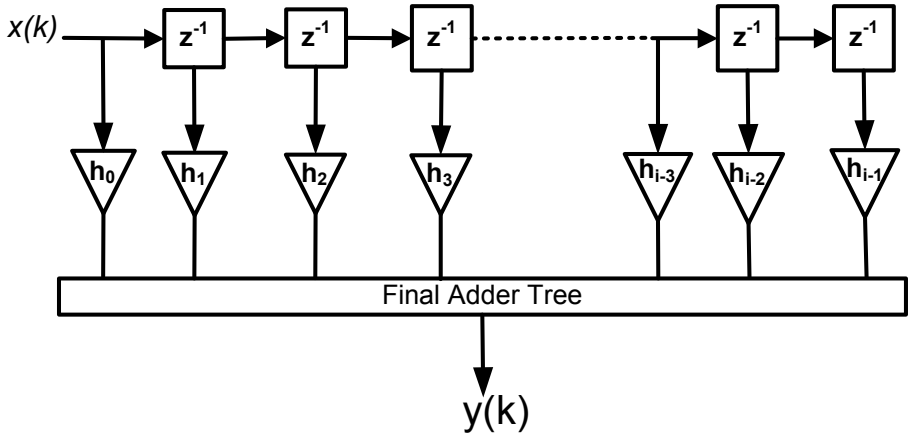


Figure 2.1 General structure of FIR filter

The general structure of a FIR filter is shown in Figure 2.1. The filter comprises a tapped delay line plus a multiply-accumulate (MAC) section. FIR filters operate such that every current sample and all previous input samples are multiplied with the coefficient (i.e., tap) values. This multiplication must take place before the next sampling instant requiring very fast multipliers that may consist of individual elements or a group of extremely fast multiplexed multiplier blocks. Mathematically, the FIR filter output *y(k)* can be described by the convolution of the filter coefficients $\{h_i \mid i = 0,1,.....N\}$ and the input signal $\{x(k)\}$ as follows:

$$y(k) = \sum_{i=0}^{N} h_i x(k-i) \tag{2.1}$$

where *N* is the order of the filter.

The operation of IIR filters is inherently recursive in nature and consequentially they exhibit a more complex structure than the FIR. Their operation is given by the recursive formula:

$$y(k) = \sum_{i=0}^{N} b_i x(k-i) - \sum_{j=1}^{M} a_i y(k-j) \tag{2.2}$$

where $\{b_i\}$ and $\{a_i\}$ are the filter coefficients. An IIR filter structure with direct form-II is shown in Figure 2.2. Unlike a FIR filter, the IIR equation contains poles. Unless the filter poles are confined within the z-domain unit circle, filter stability cannot be assured.

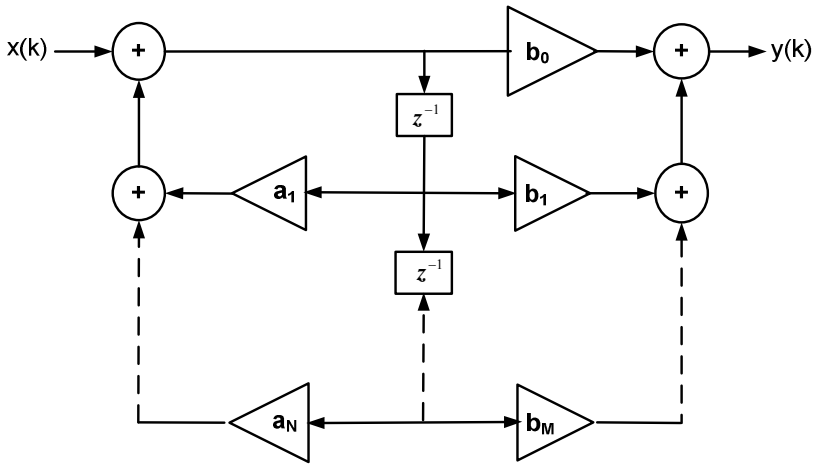


Figure 2.2 Block Diagram of an IIR direct form II filter

It is evident that FIR and IIR filter structures contains many multiplication and summation operations. When realized in integrated circuits multiplication operations typically require complex logic and a large amount of silicon area. As an example, for

a k-bit multiplication, approximately $20K^2$ transistors are required [6]. The efficiency of these traditional filters can be improved by reducing the number of transistors required for the multiplication operation. This reduction can be achieved by reducing the number of bits of both the input and its coefficients. However, simply reducing the filter coefficient or input word length will have a detrimental affect on the filtering capabilities or the output signal dynamic range.

An appropriate way to achieve this objective can be to use sigma-delta modulators that are already widely accepted for in the ADC/DAC domain. The use of Short word-length, particularly single-bit techniques derived from sigma-delta modulators greatly simplifies the arithmetic processing within filter systems. The main attraction of SWL and especially single-bit systems is their intrinsic simplicity of operation, low power consumption and stability. SWL filters can exhibit excellent area-performance tradeoffs when implemented in hardware [7].

By their very nature, short word-length systems do not require the complex integer multiplication that can be a limiting factor in contemporary multi-bit signal processing. Single-bit multiplier design can easily be implemented by simple AND/OR logic, multiplexers or small LUT blocks. This simplified design is highly attractive for hardware implementation using Field Programmable Gate Arrays (FPGA) and especially ASIC, as reducing the number of general-purpose digital multipliers in the chip is a major challenge in both these domains.

Initially, the applications of single-bit sigma delta modulated (ΣΔM) systems to mobile communications tended to be restricted to audio processing because it had proved difficult to perform complicated DSP tasks efficiently using 1-bit processing. However, in last two decades a new generation of short word-length (SWL) systems have been developed that can perform general-purpose DSP functions, including classical and adaptive LMS filtering [8-14]. The design of single-bit ternary FIR-like filter have been at the forefront of this research [10].

Ternary is a term used to describe the format of coefficients that are drawn from the set $\{+1, 0, -1\}$. In abstract analyses such as using MATLAB, the physical hardware implementation is typically ignored, but this can include conventional 2 bit binary (using three of the four available symbols) or single-line multi-level encoding [15]. Numerous ternary algorithms have been published (see section 2.3.2) that have been found to be difficult in implementation [16-18]. Various MATLAB analyses of these algorithms have been reported along with more general issues related to sigma-delta modulators such as: stability, limit cycles, chaos, idle tones, integrator spans, adaptation etc. However, the hardware implementation of bit-stream filters is rarely reported and there are still unresolved issues that need to be addressed. Some of these issues, which are investigated later in this thesis, are as follows. First of all, the efficient design of single-bit ternary FIR filters is difficult due to the requirement for high oversampling ratios that, in turn, require a large number of coefficients. Secondly, it is unclear to how to compare single-bit and multi-bit approaches in hardware as the

analysis of the relative area, power and performance of these filters tends to be a cumbersome task.

Recent rapid changes in wireless communication have increased the role of adaptive filters and channel equalizers. Many adaptive algorithms have been proposed that may be well suited to a SWL channel equalization approach [19-21]. However it is still unclear what form such a SWL adaptive channel equalizer might take and its analysis is challenging due to the single-bit nature of the adaptive coefficients that are derived after coarse quantization. In addition, rigorous stability analysis of the single-bit ternary FIR filter is still an open question that requires detailed consideration. This may lead to a modified robust design of overall single-bit ternary filters.

As a result, this thesis aims to extend the theoretical work into SWL filters to support the development of efficient ternary FIR Filter algorithms in the form of small, fast filter modules that can be used in mobile communication applications. The application of such systems can be predicted to lead to substantial reduction in hardware size and execution time. This may lead to mobile phones that are smaller, lighter, cheaper, and that run for longer on a battery charge.

This chapter presents a comprehensive survey of efficient bit-stream signal processing that encompasses sigma-delta as an integral part. This survey begins with brief introduction of short word length (often single-bit) systems and importance in the current systems. This introduction is followed by a short review of sigma-delta

modulation and its signal and noise transfer functions derivations, followed by in investigation of fast and efficient filter design algorithms based on ΣΔM. Finally, some single-bit ΣΔM are discussed along with their VLSI analysis and the literature survey is summarised.

2.2 Sigma-Delta Modulation

Oversampled sigma-delta modulators have numerous advantages over Nyquist rate conversion devices. For example, they are simple in nature, offering low cost hardware design, robust behaviour in the face of analog component imperfection and reduced complexity of the anti-aliasing filter [22]. One major advantage of ΣΔM is their inherent noise shaping that is accomplished by coarse quantization (e.g., a coarse ADC) with a feedback loop around the quantizer that suppresses the quantization noise power within frequency band of interest [23]. This important aspect of the ΣΔM that supports a good balance between bandwidth and quantization noise [24]. Hence, quantization noise is moved away from the band of interest, which allows the input signal information to be passed towards the output with minimal alteration and behaves as high pass filter for the quantization noise (or quantization spectral density).

Typically ΣΔM conversion is achieved by the oversampling ratio and noise shaping effects [25]. As a result, ΣΔMs are being proposed as alternate solutions to contemporary multi-bit signal processing designs [7, 10, 26]. Their applications are now found in diverse fields. For example, wired and wireless communication systems

[23, 27-28], DC blockers [11], arithmetic processing modules [29], neural networks [30], and audio processing, to name a few. For example, in [10] the conflicting requirements of high sampling rate, large dynamic range, and removal of the power interference before the amplification are managed by the use of sigma delta ADCs.

Typically, ΣΔM blocks are used to convert multi-bit output into single-bit format. This single-bit format generated by the sigma-delta modulators is normally filtered through a lowpass filter to reduce the quantization noise affects called demodulator and convert back the bitstream format (i.e., single-bit) into its original format. The collective filtering and down sampling operation after ΣΔM is known as decimator [25].

2.2.1 Sigma-Delta Modulator Block Diagram

The general block diagram of the sigma-delta modulator is shown in figure 1. This diagram may be divided into two parts i.e., linear and nonlinear. Loop filter is a linear part with memory element and quantizer is a nonlinear part that is without memory. The linear part is a two input system where the single output (W) can be expressed as a linear combination of its input X and S. Generally, the sigma-delta modulator loop filter has a low pass characteristic for low frequency applications such as audio processing. This low pass modulator can be manipulated to create a band pass modulator such as reported in [31].

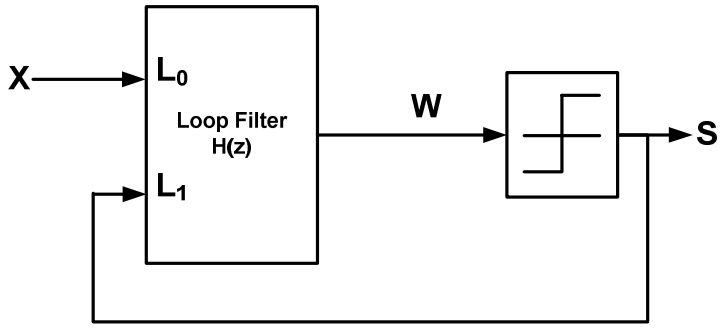


Figure 2.3 General diagram of the sigma-delta modulator

Sigma-delta modulators operate at an oversampling ratio that defines how much faster the oversampled modulator operates compared to a Nyquist-rate converter [31]. The term oversampling is a ratio between the sampling frequencies divided by the Nyquist rate that can be written as $OSR = f_s / 2f_o$. Here f_s is the sampling frequency and f_o represents the maximum input signal frequency (i.e., Nyquist criterion $2 \times f_o$). The oversampling ratio reduces the non-shaped in-band noise by 3 dB for every doubling of the sampling frequency [25]. Thus, doubling the sampling rate from f_s to f_{s1} causes the in-band quantization noise that was previously spread over $[-f_s/2, f_s/2]$ to be spread over double the frequency i.e., $[-f_{s1}/2, f_{s1}/2]$. Hence, the noise power spectral density is reduced to half its previous value. This reduction is in addition to the noise shaping affect that is an inherent part of the ΣΔM due to its built-in filtering action. This noise shaping affect may become more clear from the in-band noise spectrum approximation that is derived by considering error as white noise assumption [25, 31]:

$$Q \approx \left(\frac{\pi^{2N}}{2N+1} \right) \left(\frac{1}{OSR^{2N+1}} \right) \frac{\Delta^2}{12} \tag{2.3}$$

where Q represents the in-band quantization noise, N is the order of the modulator, and $\Delta^2/12$ is the white noise mean square value. It can easily be seen that in-band quantization noise can be reduced by increasing either or both the order of the sigma-delta modulator or the oversampling ratio. From (2.3) it is clear that the typical number of bits added to the resolution by doubling the OSR is *N+0.5* for first order sigma-delta modulators and *2N+0.5* for second order sigma-delta modulator and so on.

2.2.2 Trade-offs Between ΣΔM OSR and Modulator Order

There is a direct trade-off between the OSR and order of the sigma-delta modulator. With lower order of the sigma-delta modulators, higher OSRs are needed to suppress the in-band noise [31]. For example considering *N=2* in(2.3) i.e., second order ΣΔM, gives a decrease of in-band noise by 15dB at each doubling of OSR as compared to the 9dB that is achieved by first order ΣΔM. Thus the increase in OSR required by lower order ΣΔMs is a limiting factor that may restrict their use in broadband applications.

Three different approaches have been proposed for obtaining the better noise shaping with lower oversampling ratio [32]. One of them is to accommodate higher order of the sigma-delta modulators which gives higher noise transfer functions and reduces the noise power spectral density. The major problem with higher order of the sigma-delta modulators is their inherent instability. Employing a multi-bit quantizer is

another approach but at the cost of higher insensitivity of the quantizer itself. A third approach is to cascade the sigma-delta modulators, a technique also known as multi-stage or MASH (i.e., Multi-stage noise SHaping). Many alternate ΣΔM structures have been proposed with various orders; a good survey of these structures can be found in [31].

2.2.3 Quantizer Behaviour in ΣΔM

The overall behaviour of the sigma-delta modulators has to be considered to be non-linear due to its quantizer, thus its stability is a major concern. Normally, first and second order modulators are considered stable in nature but offer lower noise suppression. On the other hand, higher order of the sigma-delta modulators offer better in-band noise suppression but it becomes increasingly complicated to predict the quantizer behaviour.

Generally, the single-bit quantizer i.e., one that has only two possible options $\{+1,-1\}$ is preferred in ΣΔM systems due to its superior linearity compared to the multi-bit quantizer [31, 33]. However, their major problem is their higher level of quantization noise. The multi-bit quantizer has advantages over single-bit unless the quantizer does not overload, something that is less likely in higher order modulators (e.g., ≥ 3) [34].

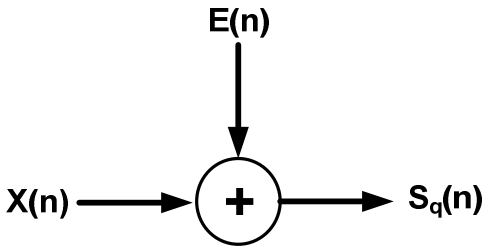


Figure 2.4 Linear Model of Quantizer

2.2.4 Linear and Non-Linear Models of ΣΔM

The design and analysis of ΣΔM can be accomplished by considering both linear and non-linear models [34-35]. The linear model of the ΣΔM quantizer is often preferred due to its greater simplicity [34]. While this linearization gives insight into the quantizer behaviour, it does not account for signal dependent quantization noise. In the linear model, the 1-bit quantizer is replaced by a quantizer variable gain that is followed by an additive white noise source with variance calculated as:

$$e_{rms}^2 = \frac{1}{\Delta}\int_{-(\Delta/2)}^{+(\Delta/2)} e^2 de = \frac{\Delta^2}{12} \tag{2.4}$$

where Δ represents the quantization interval and e represents the quantization error term (shown in Figure 2.4) that is added to the quantizer input. It states that the error is bounded in the range $\pm\Delta/2$, until and unless the quantizer is not saturated. Hence, the quantizer gain is the ratio of quantizer output voltage to the quantizer input voltage as given below:

$$\gamma = \frac{Sq(n)}{X(n)} \tag{2.5}$$

2.2.5 Sigma-Delta Modulator Z-domain Analysis

Based on a linear model in the z-domain, a ΣΔM system (see Figure 2.3) can be described by [31]:

$$W(z) = L_0(z)X(z) + L_1(z)S(z) \tag{2.6}$$

where the *S(z)* represents the quantizer output that can be described as the quantizer input plus the quantizer error signal (i.e., *E*) so that:

$$S(z) = W(z) + E(z) \tag{2.7}$$

Using these relationships, the output *S* can be written as a linear combination of two signals, namely the modulator input *X* and the quantization error *E*:

$$S(z) = STF(z)U(z) + NTF(z)E(z) \tag{2.8}$$

where,

$$NTF(z) = \frac{1}{1 - L_1(z)} \tag{2.9}$$

and

$$STF(z) = \frac{L_0(z)}{1 - L_1(z)} \tag{2.10}$$

With given desired *NTF* and *STF*, loop filter transfer function can be computed by the following relationships:

$$L_0(z) = \frac{STF(z)}{NTF(z)} \tag{2.11}$$

and

$$L_1(z) = 1 - \frac{1}{NTF(z)} \tag{2.12}$$

These relationship can be applied regardless of the structure of the loop filter and input-output characteristics of the ΣΔM are determined by solely *STF*, *NTF* and the properties of the quantizer [31]. In the simplest case, the signal is delayed by *j* clock periods in the modulator so that the STF satisfies |STF| = 1, and the *NTF* requires the quantization noise to be differentiated *N* times. Then,

$$STF(z) = z^{-j} \tag{2.13}$$

$$NTF(z) = (1 - z^{-1})^N \tag{2.14}$$

By replacing these terms in (2.10) and (2.11), we get:

$$L_0(z) = z^{-j}(1 - z^{-1})^{-N} \tag{2.15}$$

and

$$L_1(z) = 1 - (1 - z^{-1})^{-N} \tag{2.16}$$

Considering the first order ΣΔM loop filter (shown in Figure 2.5) that has a single input and only the difference $x(n) - s(n)$ enters the loop filter. Then

$L_o = H(z)$ and $L_1 = -H(z)$, and STF and NTF of the 1st order modulator are given below:

$$NTF(z) = \frac{1}{1 + H(z)} \tag{2.17}$$

$$STF(z) = \frac{H(z)}{1 + H(z)} \tag{2.18}$$

where *H(z)* is the transfer function of the common portion of the loop filter. Now, *H(z)* along with the quantizer, determines all the important properties i.e., stability, signal and noise transfer functions of the modulator [31]. From the above given NTF relationship it is evident that quantization error *Q(z)* is spectrally filtered.

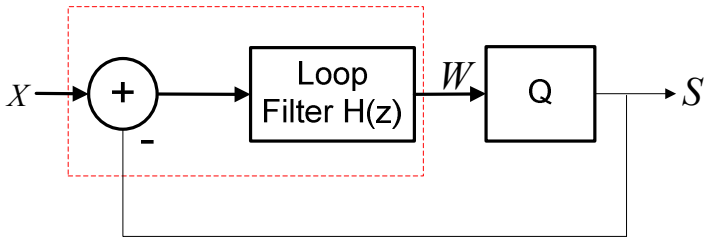


Figure 2.5 First Order Sigma-Delta Modulator Topology

To illustrate this spectral filtering let us replace the loop filter $H(z)$ by its equivalent transfer function i.e.,

$$H(z) = \frac{z^{-1}}{1 - z^{-1}} \tag{2.19}$$

where $H(z)$ is an integrator. However, more accurate models have been proposed that take into account the non-linear behaviour of the quantizer (e.g., [35]) but the model presented here (Figure 2.6) is a workable mechanism for understanding the behaviour of the ΣΔM [36]. If we now replace the $H(z)$ term in (2.17) and (2.18), then the following first order sigma-delta modulator transfer functions are achieved:

$$STF(z) = z^{-1}X(z) \tag{2.20}$$

$$NTF(z) = (1 - z^{-1})E(z) \tag{2.21}$$

where *STF(z)* is purely a delayed version of the input that does not change the form of input signal. On the other hand, the quantization noise is filtered with the differentiator $(1 - z^{-1})$ that has a high pass filter response and shapes the quantization noise away from the (low frequency) band of interest. Therefore, if the input signal is in a lower frequency range then it will be modulated in single-bit format with reduced quantization noise [23]. With a uniform distribution between the +1 and -1 for quantization noise, overall z-domain linear model that relates the output (*S*) to the input (*X*) is given as:

$$S(z) = z^{-1}X(z) + (1 - z^{-1})E(z). \tag{2.22}$$

Similarly for the N^{th} order loop filter input-output relationship can be described as:

$$S(z) = z^{-N} X(z) + (1 - z^{-1})^N E(z). \tag{2.23}$$

Here, the term N denotes the order of the modulator and $(1 - z^{-1})$ is the inherent filtering term that suppresses the in-band quantization noise. As given in (2.3), the traditional linear model for Nth-order modulator relates the output to the input spectrum according to:

$$S(e^{j\Omega}) = S_x(e^{j\Omega}) + \frac{1}{3}\left[2\sin(\frac{\Omega}{2})\right]^{2N} \tag{2.24}$$

where $S(e^{j\Omega})$ and $Sx(e^{j\Omega})$ denotes the output and input signal power spectral densities of the ΣΔM. The term $\frac{1}{3}\left[2\sin(\Omega/2)\right]$ is the squared magnitude of the NTF in the frequency band (i.e., noise spectral density) and $1/3$ is the error term with $\Delta = 2$ (i.e., $\left(\Delta^2/12\right) = 1/3$).

In general terms, quantization error is dependent on its input and is defined as difference between quantizer output and its input (i.e., $e_q(n) = q_o(n) - q_{in}(n)$). Quantization error is considered as noise when the error has statistical properties that are independent of the signal, and error samples are highly uncorrelated from sample-to-sample. Hence, the ideal in-band *SNR* (i.e., SNR_{in}) achieved by the N^{th} order ΣΔM is given below [25]:

$$SNR = 10\log_{10}(\sigma_{xy}^2) - 10\log_{10}(\sigma_{qy}^2) - 10\log_{10}\left(\frac{\pi^{2N}}{2N+1}\right) + \left(20N+10\right)\log 10(\frac{f_s}{2f_B})(dB) \tag{2.25}$$

where σ_{xy}^2 is the signal power at the output and σ_{qy}^2 is the in-band noise power at the output assuming zero mean. As the signal power is assumed to occur only in the specified signal band, it is not modified in any way, and the signal power at the output σ_{xy}^2 is the same as the input signal power σ_x^2. Thus for every doubling of the oversampling ratio (OSR), this modulator provides an extra (6*N*+3) dB of SNR [25].

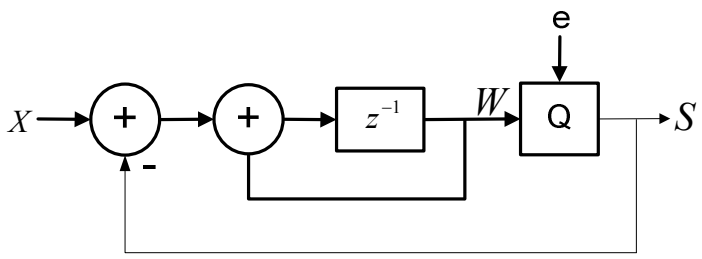


Figure 2.6 First Order Sigma-Delta Modulator Topology with Loop Filter Specified

Quantization noise can further be reduced by exploiting the noise transfer function zero locations of the typical ΣΔM topology. In [37], with fixed pole locations optimum zero locations up to 8th order of the ΣΔM are given that offers significant SNR improvement. With optimum zero locations by increasing the order of the ΣΔM (i.e., $1 \to 2 \to 3$) an additional 5-dB SNR improvement was observed in [37].

The Butterworth configuration has been the well-known choice for the pole location of the NTF. These may be described for the N^{th} order ΣΔM topology as:

$$NTF(z) = \frac{\prod_{i=1}^{N}(z - z_i)}{\prod_{i=1}^{N}(z - p_i)} = \frac{1}{1 + H(z)} \tag{2.26}$$

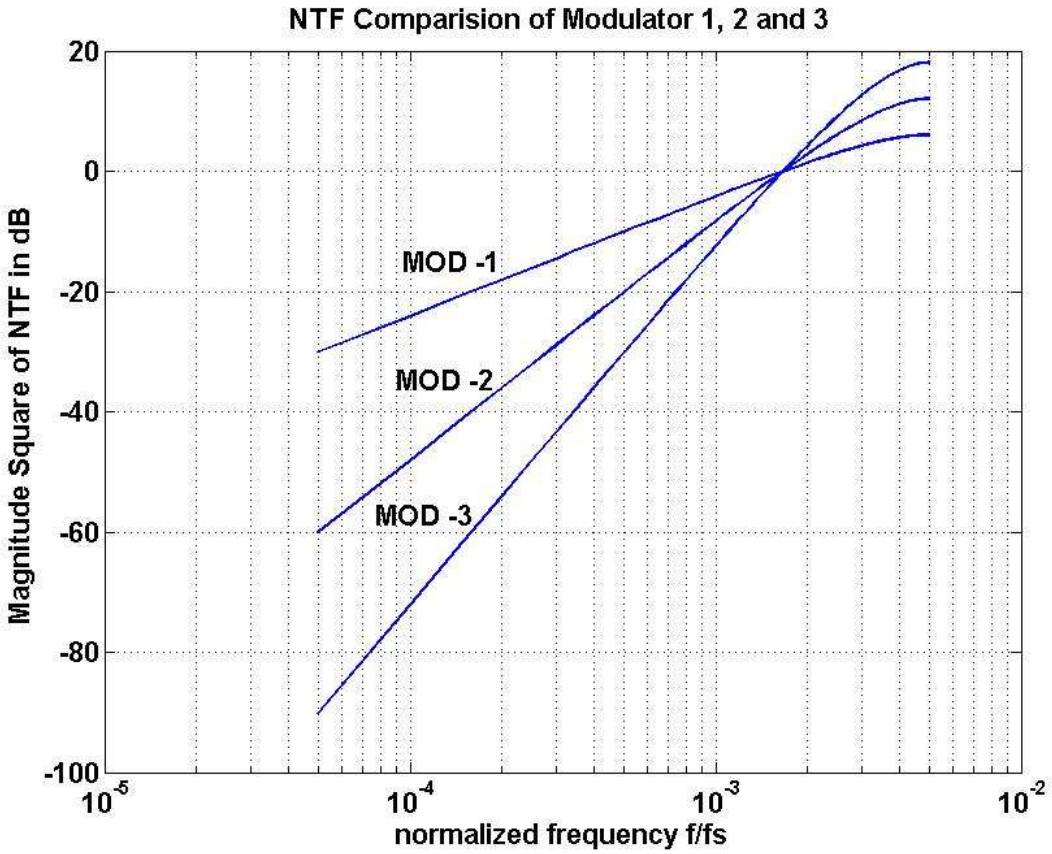


Figure 2.7 NTF at 1 – 3rd orders of the sigma-delta modulator

By increasing the order of the ΣΔM (i.e., the number of integrators) better noise suppression can be obtained. This is evident from the *NTF* relationship given in (2.26).

In order to estimate the in-band power of the quantization noise, it is useful to find the squared magnitude of NTF in the frequency domain, by setting $z = e^{j2\pi f}$. For first, second and third order sigma-delta modulators the NTF is plotted and shown in Figure 2.7. It can be seen that third order has double the stop band attenuation than second order and second order has double the stopband attenuation of the first order.

Of course, the output signal from the modulator is in a single-bit format. However, general trend is to filter that output bit-stream using traditional IIR and FIR filtering to remove the quantization noise and then to re-sample and decimate to the Nyquist frequency. Hence, once the ΣΔM output is filtered using one of these traditional techniques, the signal is again in a multi-bit format. However, the aim of SWL DSP techniques is to avoid, as far as possible, multi-bit stage(s) throughout the system design using ΣΔM.

2.3 Fast and Efficient FIR Filter Design Techniques

It is no surprise that many signal processing tasks can be accomplished by a microprocessor or a digital signal processor (commonly called DSP kits). Built-in multiplication modules are the core element of these devices. Furthermore, implementation of multiply and accumulate (MAC) circuits within signal processors can significantly improves the throughput of FIR and IIR digital filters structures (see Figure 2.1 and Figure 2.2) that requires large number of multiply and accumulation operations per sampling period.

An alternative solution is to use gate-level programmable devices such as field programmable gate arrays (FPGAs) to perform the digital filtering tasks. Concurrent (i.e. parallel) mode of operations of these devices is of great interest as it can improve the throughput of the digital signal processing circuits especially digital filtering modules. This higher throughput can be achieved at the cost of higher chip area compared to the serial implementation of the circuits. Nonetheless, many of these FPGA devices also possess higher number of built-in multipliers that requires large amount of silicon space within in the FPGA. The most recent FPGA devices have include resources that easily support general purpose signal processing tasks even within mid-range commercial devices.

However, there is direct trade-off between chip area and throughput in these devices. Some obvious applications that require fast and efficient digital filters are decimation filters, audio filter banks, charge-coupled-device filters and software defined radio, all of which require high throughput. To achieve fast and efficient implementations, many techniques have been proposed. The overarching theme of these techniques has been to reduce the complexity of the multiplication process in any way possible. One method of reducing the complexity of the multiplier is to reduce the word length in both input and the filter coefficients. The preferred approach is to utilize the sigma-delta modulation to reduce the word length; this thesis will focus on these methods. Here are many techniques that use some form of sigma-delta

modulation or the like to improve the efficiency of the digital filtering operations. Examples of such techniques were reported in [26, 36, 38-44].

2.3.1 Fast FIR Filters

Fast and efficient filters generally fall in two classes: sigma-delta modulation based and optimization techniques within a multi-bit format. A brief description of both these methods is given below.

2.3.2 Sigma-delta Modulation Based Fast Filters

Much work has been reported on the design and implementation of the sigma-delta modulation based FIR and IIR filters encompassing various forms. The work that was commenced by [16], and progressed by [43, 45] has been reported by many, such as [17, 24, 36, 38-40, 42, 46-47]. More recently sigma-delta modulation based bit-stream adder and multiplier modules have been described in [48-49].

In [40, 42, 46], the efficient FPGA implementation of a narrowband FIR filter is achieved by simplifying the MAC operation using a lower precision input to the filter. This filtering operation requires that the input to the filter should be oversampled and re-quantized through the error feedback ΣΔM as shown in Figure 2.8. Using Authors have used distributed arithmetic (DA) approach to design error prediction FIR filter that has been placed in negative feedback path (see Figure 2.8). This prediction filter has a flat pass band and leading phase shift in the band of interest. The paper also discusses the optimum prediction filter design based on statistics and a minimum-

mean-squared error (MMSE) calculation. For FIR filter input, only 3 – 4 bits in the re-quantizer output were processed as compared to the original 16-bit input bit-stream to the ΣΔM (i.e., re-quantizer).

Overall, an efficient implementation of a narrowband digital filter through a re-quantizing operation has shown a 50% reduction in logic resources as compared to a traditional FIR filter implementation using a FPGA. This filter shows a great promise for FIR filter implementation. Further reduction in complexity can be gained through harsher requantization to lower precision words.

In [45] and [43] fast and efficient FIR filters are presented. The authors discussed two sigma-delta filtering approaches. In first approach, FIR filter coefficients are encoded using first order sigma-delta modulator. Hence, the input to the filter must be interpolated and zero-padded to R times sampling frequency. An efficient two step interpolation process was proposed that required firstly interpolating the original signal (x_n) by 4 times at a sampling frequency of $4f_N$ resulting in x_{of}. This signal (x_{of}) was then up-sampled by $R/4$ times (R is the typical OSR) to give $\hat{x}_n$ by appending zeros. The proposed structure is shown in Figure 2.9. The decoder for this filter is used to reconstruct the original signal by resampling to the Nyquist rate and removing quantization noise by using a low pass filter and decimator.

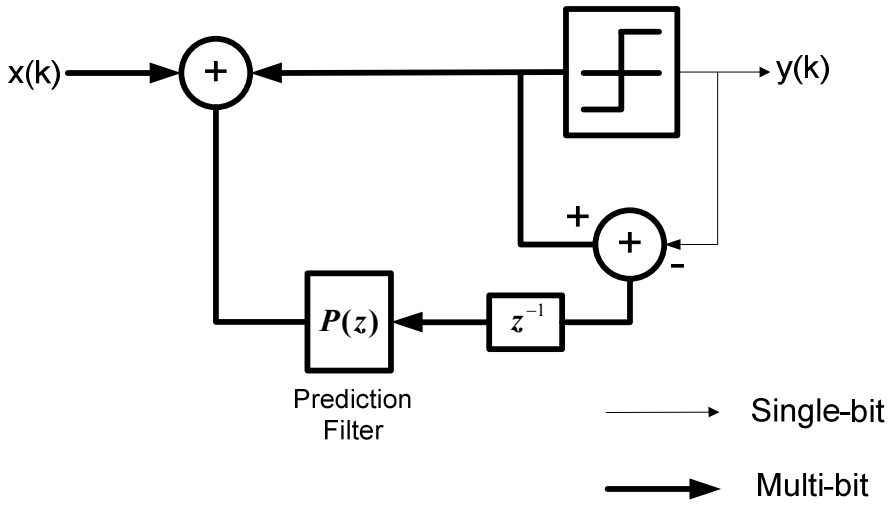


Figure 2.8 Block diagram of the error feedback ΣΔM for requantization

The use of cascaded comb filters as reported in [50] was adopted to further simplify the decoder design whilst removing any alias introduced into the system from the FIR filter. Only two cascaded comb filters were used in design (shown in Figure 2.10) because the authors found that using more than two cascaded comb filters did not improve the trade-off between signal-to-noise ratio of the coded output and the OSR.

In a second approach (Figure 2.11), input data was encoded into single-bit format through sigma-delta modulator whereas the filter coefficients were kept in PCM format. The decoder for this structure was identical to the one shown in Figure 2.10. Signal encoding with sigma-delta modulator worked as an ADC and single-bit coder so there is no longer a requirement for a conventional ADC. Further, no input interpolation is required in this setup as the signal passing through ΣΔM will be oversampled.

To perform the filtering operation, full precision filter coefficients were zero padded by R to match the oversampling ratio of the ΣΔM. Decoder circuits comprising cascaded comb and baseband filters were used to remove the quantization noise and aliases from the filtered output signal. However, the output signal was in multi-bit format in these schemes.

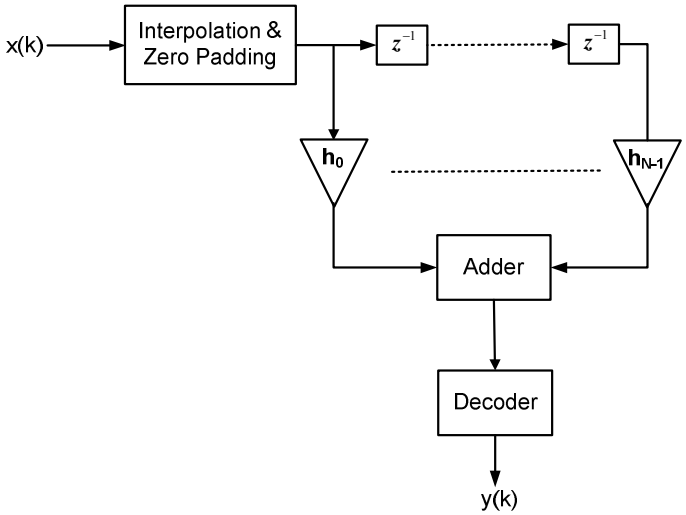


Figure 2.9 Block diagram of the FIR filter with ΣΔM modulated filter coefficients

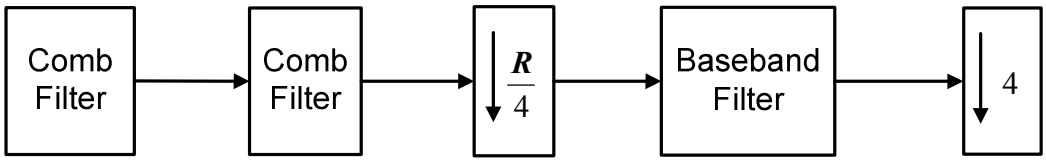


Figure 2.10 Block diagram of the decoder used in FIR filter with ΣΔM modulated filter coefficients and with ΣΔM modulated input signal

In [43] the authors also propose a fully sigma-delta modulated FIR filter. In this instance, it was recognized that filter performs well if both the input and filter

coefficients are sigma-delta encoded in single-bit format. A similar structure was utilized to that shown in Figure 2.9 except that the interpolator was replaced with a sigma-delta modulator. It was found using simulation that the design exhibits a flat input spectrum in the Nyquist frequency range and the latter approach (Figure 2.11) performed well in comparison to the former (Figure 2.10). This structure was found to further reduce the complexity of the filter's hardware implementation.

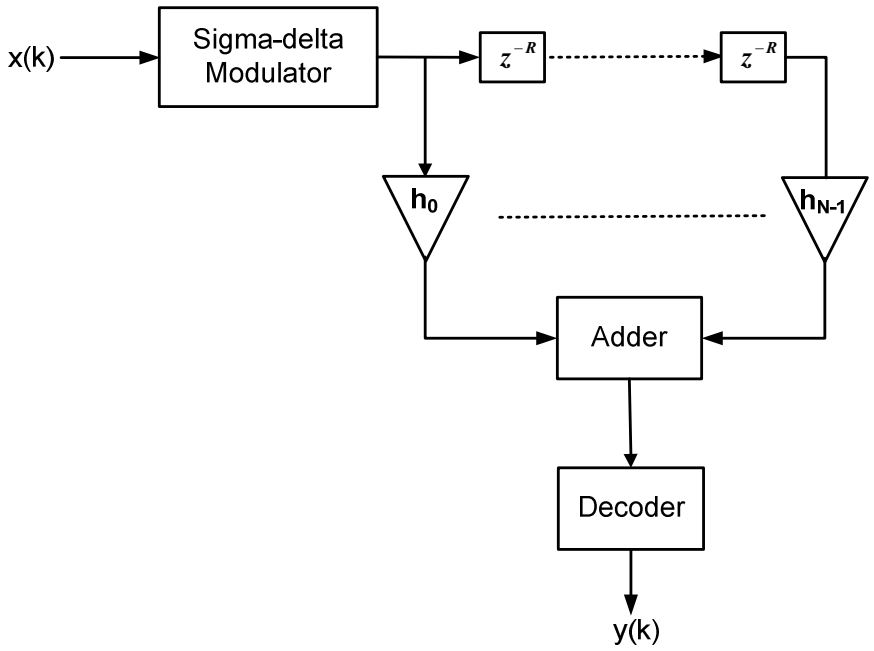


Figure 2.11 Block diagram of the FIR filter with ΣΔM modulated input signal

A ternary format has an extra symbol for input and filter coefficients and has been found to offer better stop band attenuation and dynamic range flexibility compared to the binary format [7, 43]. While that work illustrates the potential benefits of ternary

encoded filters, the final decoder leaves the output in a multi-bit format that again requires complex hardware to process.

A slightly different fast and efficient FIR filter design using sigma-delta encoding is presented in [17] in which a Look-Ahead Decision Feedback (LADF) approach is used to encode the filter coefficients into a single-bit format. In that work, the proposed technique is compared with two other ΣΔM architectures: the multi stage (MASH) and double loop (DSM2). It was found that the proposed architecture outperforms in comparison to double loop but has poor performance against the MASH architecture. Given the lower complexity implementation of the proposed architecture, the author argues that the method is appropriate for filter encoding. However, the quantizer stage with LADF architecture is more complex than the single-bit quantizer and its associated ΣΔM architecture.

The last group of fast and efficient filters designs use a canonical signed digit (CSD) quantizer with signed powers of two ΣΔM output [47]. It is argued that the CSD quantizer provides many more quantization levels than a single-bit quantizer, which suits the linear modelling of the system design and can improve the system stability. Thus an output in CSD format obtained from the ΣΔM can be used as the FIR filter coefficients and the multiplication operation becomes simple shifts. Another, promising scheme, presented in [51] uses a slightly more complex architecture but is essentially the same technique.

2.4 Single-bit Filtering Techniques

Regardless of the many optimizations that have been proposed, a large number of multiplication stages still translates into large area, delay and power consumption. One-bit ΣΔ modulators are widely used in AD and DA conversion stages due to their inherent linearity and precision. However, it is less common for the entire digital processing path to operate on single bit data. The more usual approach has been to decimate the signal data stream after conversion and for the remaining processing to be performed in standard binary at the Nyquist rate and with a resolution mandated by dynamic range and noise considerations.

Sigma Delta Modulation (ΣΔM) encoding of the FIR filter coefficients has shown to be efficient way to reduce the complexity of the multiplier and improve its area–performance tradeoffs [52]. The simple arithmetic of single-bit DSP systems results in efficient hardware implementations that map well to FPGA resources, which comprise flip-flops plus simple logic blocks and/or look-up tables. The advantages of single-bit systems were first identified by [16] and further developed in [45, 53] and [47]. Recently, general purpose Short Word Length (SWL) DSP applications including classical LMS algorithms have been described in [10, 14]. In this section we introduce and describe the techniques that have been used to filter whilst maintaining a single-bit output. This section is further divided into two sections i.e., simulation based single-bit techniques analysis and its VLSI analysis.

2.4.1 Single-bit techniques

As the name suggests the single-bit filters produce single-bit output. In last decade various general purpose DSP applications are reported using single-bit sigma-delta modulation encoding including classical FIR filter in [10-11, 14, 54-55]. This single-bit approach was first reported for IIR and FIR filtering in [39] and [36].

In [36], single-bit FIR filtering technique is proposed with bit-stream input and fixed or floating point coefficients similar to the one reported in [43, 45]. However, the major contribution is the replacement of the decoder in [43] by a ΣΔM that has a low pass signal transfer function. The single-bit FIR filter as proposed in [36] is shown in Figure 2.12.

Similarly, in the second approach presented in [45], the input is assumed to be in single-bit format, while full precision filter coefficients are generated at the Nyquist rate. This newly generated impulse response was interpolated by R times, where R is the oversampling ratio of the input signal, via zero-interleaving. The R aliases that were introduced due to zero-interleaving in [45], were removed by decoder comprising of cascaded comb filters and baseband filter. However, in [36], ΣΔM was used instead of a decoder. This ΣΔM was used to remove the aliasing created by the zero-interleaving process and served to re-modulate the multi-bit output signal from the FIR filter back into the single-bit domain.

The VLSI analysis of the proposed design was carried out and authors found that single-bit design to be more efficient in silicon resources than a PCM digital filter up to 80 taps. The structure still has the complexity of a full precision filter coefficients, this can also increase the word length of the FIR filter output.

The re-modulator complexity is discussed by the same authors in [56]. Digital ΣΔM low pass frequency responses are typically not easy to find in the current literature. A fourth order ΣΔM was used for this purpose with various powers of two multiplications that created more complex SDM structure than standard one. Therefore, low pass modulator structure presented in [56] is very complex for single-bit filters.

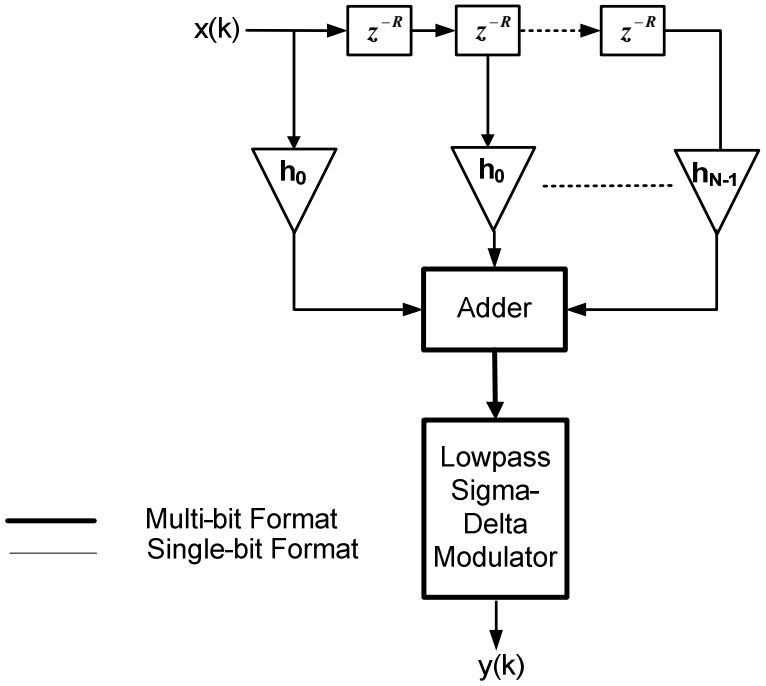


Figure 2.12 Block diagram of the single-bit FIR filter

The core idea of the IIR single-bit ΣΔM presented in [6, 39] was to multiply one-bit oversampled input signal with a multi-bit fixed coefficient. The resulting multi-bit output must be applied to the sigma-delta modulator to get back the single-bit output. Initially the model was tested without the feed-forward integrator which resulted in a large noise gain due to higher oversampling ratio transfer function. A modified version with an integrator inside the loop that resulted noise reduction and keeping the STF and NTF same is shown in Figure 2.13. In this model, the ΣΔM is assumed to be a single delay element, hence, the system is a basic first-order recursive filter [6, 39].

The stability of the system in Figure 2.13 was assumed to be determined by a rule of thumb with an assumption that second order sigma-delta modulators will remain stable. But due to an extra integrator inside the loop the overall NTF becomes equivalent to a third order ΣΔM, making it more difficult to analyse the stability of the overall system [39]. Therefore, this system was not further studied by those authors.

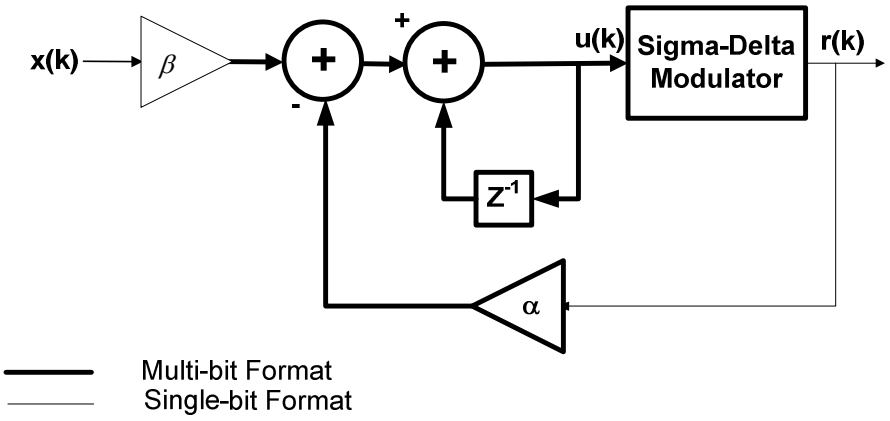


Figure 2.13 Block diagram of the first order single-bit IIR filter

However, a quasi-orthonormal state space IIR architecture was shown to have good filtering abilities with good stop band attenuation by the same authors in [57]. The downside of this structure is that it requires N ΣΔM blocks for an Nth - order IIR filter and the structure becomes very complex as the number of order increases. The proliferation of ΣΔM blocks only adds to the quantization noise in the band of interest and makes any stability analysis very difficult [39].

Recently, new DSP design techniques called short word length (SWL) have been reported in [5, 10-11, 13]. Of these SWL techniques, the so-called single-bit ternary FIR filter was first proposed in [10]. This design is comprised of two parts: the ternary filter and the IIR remodulator as shown in Figure 2.14. A new method to generate the single-bit ternary filter was also proposed that starts with the selection of the target impulse response. This target impulse response must undergo an interpolation stage before the ternary sigma-delta modulated form of the filter can be generated. The generated ternary format of coefficients must have flat pass band frequency response in the frequency band of interest (i.e. $0 \rightarrow f_0$). The transfer function of the overall design was derived and the filter was simulated at a number of OSR values. It was found that the resulting single-bit filter produced an equivalent output to the target impulse response. Hence, it appears that single-bit ternary filters can take over the bulky multi-bit systems that include complex multiplication.

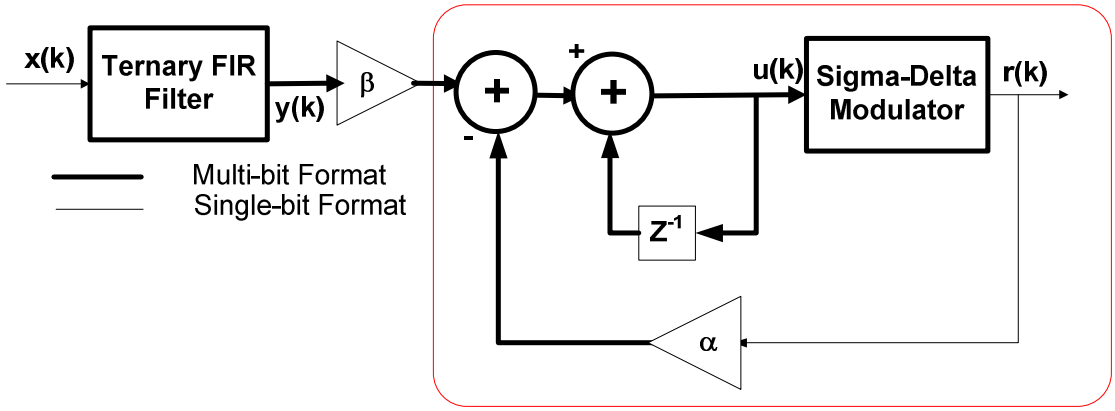


Figure 2.14 General bock diagram of the single-bit ternary FIR filter

Using the same approach, a narrowband band pass ΣΔM was proposed in [13] (Figure 2.15). Again it comprises two parts: the ternary filter followed by the re-modulation of the multi-bit into single-bit format. Unlike low pass single-bit filter, these authors have proposed a re-modulation by a simple band pass ΣΔM that has efficient architecture and less stability sensitivity compared to the IIR re-modulator. Coefficients were encoded into ternary format by passing the band-pass target impulse response through an 8^{th} order ΣΔM with optimum coefficients. Through MATLAB simulation it was found that the overall frequency response of the proposed method as was very similar to the original target impulse response.

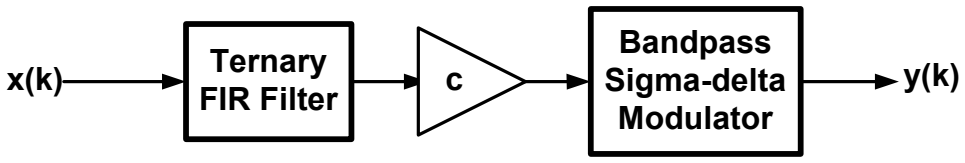


Figure 2.15 Single-bit narrowband bandpass FIR filter

The performance of the proposed method is also discussed in [58]. It was found that FFT and spline interpolation techniques offer superior stop band attenuation performance to other techniques. Following the same approach, single-bit resonators and BFSK demodulator designs have been reported in [28, 55]. However, this short word length approach was not verified through hardware synthesis nor was its area-performance-power compared with contemporary multi-bit techniques. Furthermore, that work does not extend to a rigorous stability analysis of the SWL filtering techniques.

Further to this work, a LMS-like single-bit adaptive filtering structure for noise cancelling has been presented in [5, 14, 59], in which all input, output, and filter coefficients are in single-bit format. Overall, three short-word length adaptive structures were proposed: namely, ternary, single-bit and 2-bit. The overall weight vector equation was derived by using block LMS algorithm which has advantage of accommodating more data samples and better performance than a sample-by-sample LMS algorithm. Through MATLAB simulation it was found that 2-bit single-bit adaptive filter has superior performance than others i.e., single-bit and ternary at the cost of prospect more chip area.

However, much work is still needed to explore the design using random input with higher noise environment. In addition, it is still unclear what might be the optimum coefficient update rate or range of the convergence parameter (*mu*) or shape of the learning curves.

2.5 VLSI Analysis of ΣΔM based bit-stream circuits

Although much work has been reported on the design and analysis of single-bit systems, it appears that there has been little reported on rigorous hardware analysis of single-bit signal processing techniques using FPGAs. However, we could find very small work reported on VLSI synthesize and analysis of bit-stream arithmetic modules and its variants that are reported below. However, these arithmetic modules are building blocks of the DSP algorithms but not a signal processing application itself. Furthermore, these modules have been an inherent part of the already proposed single-bit systems in [10, 14].

In [26, 60] efficient bit-stream (i.e., single-bit) arithmetic modules are presented for mobile communication in which general purpose modules including adder, multiplier, divider and square root have been designed. A typical QPSK communication model has been demonstrated by using the proposed bit-stream arithmetic modules and a 40% reduction in logic gate count compared to conventional design has been reported.

Bit-stream arithmetic modules with bi, tri, and quad level are reported in [7, 49, 61-62]. In these cases, hardware implementation of the arithmetic modules is done in Xilinx Virtex-5 using two's complement representation. Through synthesize results, it was found that there is significant improvement in the signal-to-noise ratio and performance with ternary format than binary at the cost of a more complex structure [7].

Bit-stream (i.e., single-bit) ternary and multi-bit approaches have been compared in FPGAs by synthesizing a Type I digital phase locked loop (DPLL) application using a direct digital synthesis approach [63]. Bit-stream ternary approach was found resources efficient than its corresponding multi-bit system [7].

In [61-62], an efficient implementation of bit-stream adders and multipliers modules is reported in FPGAs. In [61], a (4,2) adder structure (i.e., 6-input (4+2)) was exploited that better suited the Altera and Xilinx 6-input LUT architectures than a conventional (2,1) architecture. The proposed adder structure resulted in a 50% reduction in LUT count and a 20% higher clock frequency [61]. In [62], the tri level bit-stream was extended to quad level and compared to the sorter based approach [64]. The quad-level bit-stream adder and multiplier presented in [62] were encoded using 2-bit and the truncated third bit was fed back to the adder to suppress the truncation error. Through Xilinx FPGA synthesis, the proposed adder and multiplier have shown significant improvement in area-performance compared to the sorter approach [64]. This quad-level adder approach resulted in about a 76% LUT reduction and a 93% higher clock rate, while the proposed bit-stream multiplier showed a 82% LUT reduction and 122% higher clock frequency [62].

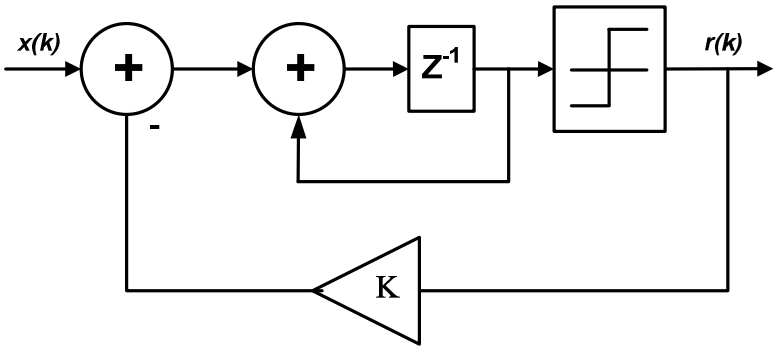


Figure 2.16 First order digital sigma-delta modulator[49]

Regardless of the work reported on simple arithmetic modules, the drawback to all of these reported works is the limited range of their adder and multiplier modules (i.e., $L = 4$). Further, there has been no detailed comparison provided to its corresponding multi-bit system except for one example reported in [7]. From the general behavior of multi-bit versus single-bit systems, it can be predicted that there may a cross-over point between the two approaches, where one could be preferred over the other. This idea will be addressed later in the thesis.

Unlike [7, 61-62], in [49], an existing IIR low pass filter ΣΔM is utilized to generate the input bit-stream and the design has been analysed to characterise the selection of the ΣΔM design parameter (i.e., K, shown in Figure 2.16) . The design was extended to tri and quad level and compared with the bi-level ΣΔM. The noise performance was simulated for all three types of multipliers and it was found that tri-level design has a performance gain of 11.2 dB over the bi-level design, while the quad-level design has about a further 8.8 dB gain over the tri-level design [49].

Hardware implementation of all three designs was performed using the Xilinx Virtex – 5 FPGA and the area-performance characteristics of the multiplier were noted. The synthesis results show a direct trade-off between all three designs and the two approaches for the IIR and FIR filter modules. These results indicate that the bi-level design is more resource efficient than either of the tri and quad level and provides higher performance at the cost of lower noise suppression and vice versa. However, this assumes that the system was stable by considering the same approach described in [39]. This rule of thumb may lead to inappropriate solution in real systems [39].

2.5.1 FIR Filter Design Techniques in FPGAs

This section describes input and coefficient encoding techniques that can be exploited to implement the fast and efficient DSP algorithms in FPGAs. The techniques can be applied in either a single-bit or multi-bit environment [10, 13].

As outlined above, it is the performance of the multiply-accumulate (MAC) stages that will have the greatest impact on the overall behaviour of digital filters, including FIR types. Thus, various filter design techniques have been proposed that specifically target the complexity of these stages. For example, distributed arithmetic is a common technique that has been used in FPGA designs for many years [65], [66] in which the multiplication stages are performed using Look-up Tables (LUTs), thereby reducing the overall size of the hardware. In [2] Systolic Distributed Arithmetic was used by to improve the area-performance-power tradeoffs of a FIR filter design implemented on a Xilinx Virtex-E device at various filter orders but with a fixed coefficient bit-width

(i.e. L=8). It was observed that the best tradeoffs between area-performance and power can be achieved at an address length of four.

Many other techniques have been proposed: Canonical Sign Digit (CSD) [67]; the Dempster Method [68]; Mirror Symmetric Filter Pairs [4]; two-stage parallelism [3] and Redundant Binary Schemes [69] to name just a few. Methods specifically aimed at FPGA-based FIR filter implementations include the fully pipelined and full-parallel transposed form [70], Add-and-Shift method with advanced calculation [71] and hardware efficient distributed arithmetic for higher orders [66], [65]. In [3], a new design technique based on a linear phase prototype filter that exploits coefficient symmetry was shown to offer better performance at a hardware cost similar to that of linear phase filters. Further, [3] also described a transpose direct-form with CSD multipliers that offers better area-performance tradeoffs when using classical methods.

Apart from the classical multiplier complexity reduction techniques, a new approach called Slice Reduction Graphs (SRG) [70], which reduces area by minimizing the multiplier block logic depth and pipeline registers, has been shown to offer improved area-performance over the Reduced Adder Graph (RAG) and Distributed Arithmetic (DA) techniques. In [70], simulations were carried out at coefficient bit-widths in the range of 2–20 bits, while keeping the order of the filter constant (i.e., at 51). The order of the filter was then varied in the range 10–250 at fixed coefficient bit-widths. The maximum average operating frequency achieved by

the proposed technique was in the range of 175–180MHz at the lowest filter order, further reducing towards 150–160MHz as the filter order increased above 60.

The primary intent of the techniques mentioned above has been to improve the area-performance characteristics of parallel multi-bit binary filters operating at the Nyquist rate. However it is obvious that the format of the coefficients and input data is one reason for the high complexity of the MAC stages. In [45, 53, 72], the complexity of the filter coefficients has been addressed by employing a simple single-bit coefficient format. This technique can reduce the hardware complexity of multipliers to simple AND-OR logic or small look-up table (LUT) organizations.

2.6 The Role of FPGAs in Mobile Communication

Use of wireless communication has increased exponentially in the last decade and have reached nearly to 6 billion subscribers. The key challenges for wireless communication have been high data rate, low latency, high reliability, efficient hardware implementation, optimum cost-benefit tradeoffs and the development of more efficient modulation schemes [73]. These challenges have been variously addressed by 4G technologies such as Long Term Evolution (LTE), Advanced LTE and WiMAX and by employing Multiple Input Multiple Output (MIMO) antennas and efficient modulation scheme, particularly orthogonal frequency division multiplexing (OFDM) [73].

As a result, wireless communication depends greatly on the efficient hardware implementation of advanced MIMO-OFDM receiver architectures required to support 4G technologies [74-75]. For example, [73] has shown that an approach based on Field programmable gate arrays to address the architectural challenges associated with spatial multiplexing. Further, the choice of FPGA technology allows for flexible architectures supporting spatial multiplexing MIMO detector, Flex-sphere, and beam-forming within a WiMAX system. The FPGA approach is highly useful due to its inherently parallel nature, reconfigurability, shorter time-to-market, and the availability of advanced DSP IP cores for an efficient hardware implementation [73-74, 76-77].

In [74], FPGA-based communication receivers for smart antenna array embedded system have been implemented on Altera Stratix technology FPGA. This work has compared single and multi-branch FPGA-based receiver designs in terms of error rate performance and power consumption. It is concluded that the flexibility and high on-chip resources available on an FPGA make it very useful for adaptive receivers for future wireless generations.

Similarly, in [78-79] it has been shown that FPGAs are of interest in the development of software defined radios due to its reconfigurable nature and provision of advanced signal processing algorithms. In [79], a 16QAM-based software defined radio has been implemented using Xilinx FPGA. It is shown that the proposed model

achieves minimum BER while operating across a range of different standards like; GSM, OFDM, and WCDMA.

In [80], various targeted design platforms has been introduced for enabling wireless communication by FPGAs. For example – Multi-mode Radio Targeted Design Platform has been designed for high performance at low cost. The platform can be reconfigured to support any of the major commercial air interfaces, including LTE, WiMAX, W-CDMA, TD-SCDMA, CDMA2000 and MC-GSM. It has also been reported that LTE baseband targeted design platform in conjunction with the Multi-mode radio targeted design platform supports the creation of end-to-end LTE base-station design encompassing radio, baseband, media access control (MAC) and transport functions. FPGA based simplified modular platforms increases performance and reduces cost for frequency division duplex (FDD) and time division duplex (TDD) [80].

2.7 Summary

In this survey we have described previous work on the development of fast and efficient filter designs, on sigma-delta modulation, its signal and noise transfer function, single-bit schemes to design an efficient signal processing applications and finally on efficient filter input and coefficient encoding techniques.

Single-bit design techniques have been studied since the early 80s, having been first reported by [18] and further enhanced by [43, 45] and in [6, 39]. Recently, new single-

bit signal processing techniques have been introduced. Known in general as the Short Word Length (SWL) approach it is more suitable to hardware implementations, especially FPGAs.

Though significant work has been reported on single-bit design techniques, few analysis of the VLSI bit-stream circuits appear in the literature. The design, analysis and FPGA synthesis of arithmetic modules (i.e., adder, multiplier and divider) were first reported in [60] and further advanced by in [7, 49, 61-62]. By synthesizing to the Xilinx Virtex-5, significant improvement in logic reduction was found by using tri, and quad level for the arithmetic modules.

This thesis aims to explore the design and synthesis of SWL DSP algorithms in FPGA and to analyse important characteristics such as area , performance and power tradeoffs. In the following chapters we present the comparison of SWL FIR-like filter with its multi-bit counterpart. In a second stage, three encoding techniques have been analysed to determine their effect on the area and performance of SWL applications.

The necessary stability criteria for this SWL FIR-like filter have also been re-visited and an improved model proposed that offers more control over the internal loop. Finally we have designed a new mathematical model for single-bit adaptive channel equalization and simulated it in MATLAB.

Chapter – 3

Power-Area-Performance Characteristics of FPGA-based ΣΔ FIR Filters

3.1 Introduction

As described in Chapter 2, Sigma Delta Modulation (ΣΔM) are widely used in AD and DA conversion stages due to their inherent linearity and precision. However, it is less common for the entire digital processing path to operate on single bit data. The more usual approach has been to decimate the signal data stream after conversion and for the remaining processing to be performed in standard binary at the Nyquist rate and with a resolution mandated by dynamic range and noise considerations.

It is also clear that ΣΔM encoding of the FIR filter coefficients represents an efficient way to reduce the complexity of the multiplier stage and improve its area–performance tradeoffs [52]. The simple arithmetic of single-bit DSP systems can result in efficient hardware implementations that map well to FPGA resources, which comprise flip-flops plus simple logic blocks and/or look-up tables. The advantages of

single-bit systems were first identified by [72] and further developed in [45, 53] and [47]. Recently, general purpose Short Word Length (SWL) DSP applications including classical LMS algorithms have been described in [10, 14].

Although the general structure of a single-bit filter is similar to its multi-bit counterpart, it has to operate at a large Over Sampling Ratio (OSR) in order to achieve an equivalent level of performance. The order of a single-bit filter is directly related to the order of the OSR. Increasing the OSR increases the order of the filter and improves the performance at the expense of more hardware. At the same time, a high operating frequency is required to achieve a given level of performance, so its implementation becomes more challenging.

While it is true that single-bit ΣΔM DSP systems have tended to be applied to ADC and audio processing applications, recent results (e.g., [81]) have shown that they can be operated at clock speeds in excess of 400MHz and with a dynamic range beyond 70dB, making them suitable for video processing applications as well. Nevertheless, it is still not immediately clear whether the use of ΣΔ modulated coefficients on short word-length data (i.e., binary or ternary) will necessarily result in smaller or more power-efficient filter designs.

In this chapter a novel design of a single-bit ternary FIR-like filter is proposed and implemented in FPGA. Area, power and performance comparisons are also presented

for a range of single-bit and multi-bit FIR filter designs with equivalent spectral performance. Much of the material in this chapter has been published in [82-84]

3.2 FIR Filter Design Techniques

As outlined above, it is the performance of the multiply-accumulate (MAC) stages that will have the greatest impact on the overall behaviour of digital filters, including FIR types. Thus, various filter design techniques have been proposed that directly address the complexity of that stage, many of which have been described in Chapter 2.

A common theme amongst the techniques so far described has been the need to overcome the complexity inherent in the processing of multi-bit arithmetic. An obvious potential solution is to employ a short word length approach to eliminate the main source of this complexity i.e., the multiplication stage. Taken to its minimum extreme, a two-input, single-bit multiplier becomes a simple AND gate. A two-input, two-bit multiplier is not much more complex. However, the associated requirement to run at a significantly higher sampling rate might conceivable negate whatever advantages are gained from the reduction in bit width. Thus, it is not immediately obvious that SWL methods offer any compelling advantage over conventional techniques. We will examine this issue in this and the following chapter.

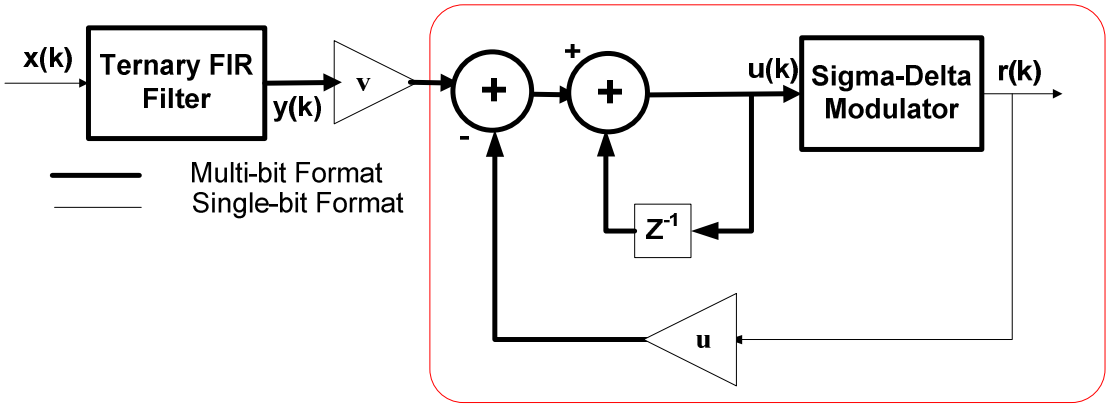


Figure 3.1 General Block Diagram of Single-bit FIR filter structure (adapted from [10]).

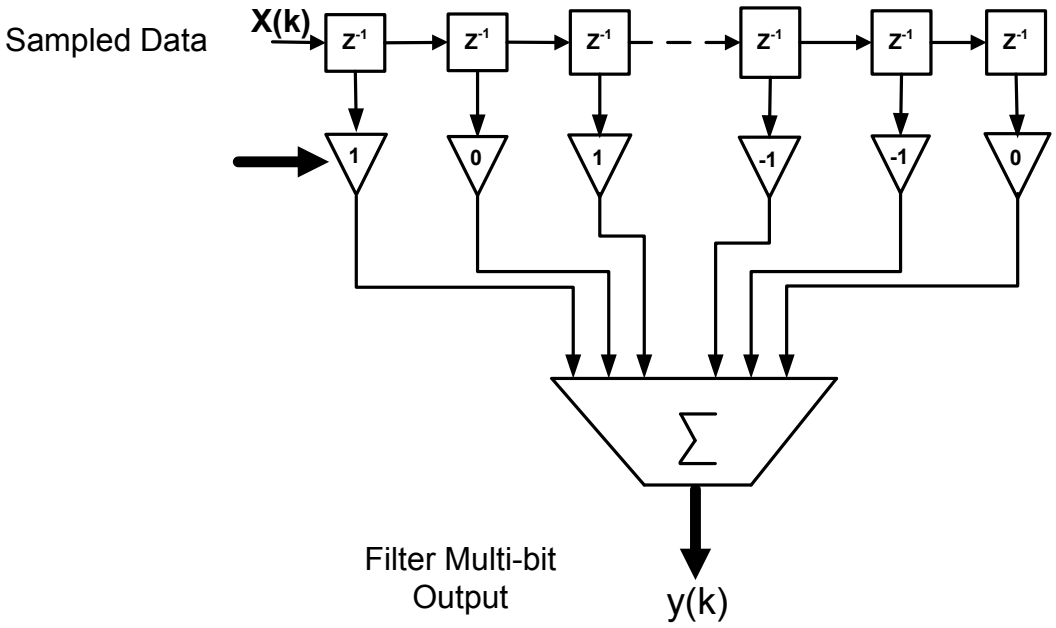


Figure 3.2 Block diagram of Ternary FIR filter (adapted from [10])

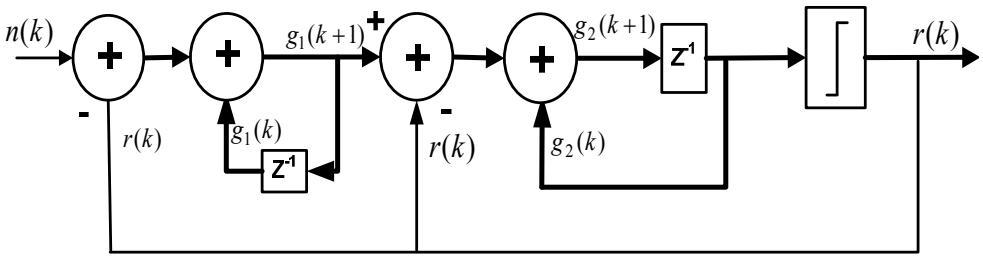


Figure 3.3 Second Order ΣΔM architecture

Of course, the issue of multiplier complexity can always be addressed by assuming constant coefficient multiplication. Whereas a full multiplier can handle any arbitrary combinations of two multiplicands, if one of the multiplicands is a constant, a far more efficient implementation will involve simple look-up tables plus adder/subtractor modules [85]. Clearly, the actual complexity of these multiplier structures will depend entirely on the value of the constant. Further, if the coefficient is zero, the multiplication step can be removed completely. Although it is difficult to generalize, in the work leading to this thesis it has been observed that in the order of 30% of coefficients, across a range of FIR filter designs, tend to be zero. Obviously, this simplification is available to single bit filters as well. In this case, multiplication by $\{+1, -1\}$ can be replaced by a simple wiring connection to $\{\overline{D_i}, D_i\}$, while a coefficient of zero implies that the connection can be removed completely, along with any part of the "accumulate" function to which it connects.

However, this work is intended to lead towards the implementation of adaptive filter implementations in which the coefficient values change over time. In this case, as

in many similar applications, fixed coefficient optimisations of this type are not available. Thus the following analysis will assume that comparisons are being made between full implementations of the competing filter organizations.

3.3 Single-Bit Ternary FIR-like filter

In this section we examine the architecture of a single-bit Ternary FIR-like filter (SBTFF) designed using a direct-form structure with ternary coefficients (Figure 3.1) [10]. Here, a *Ternary FIR* filter is defined as one in which the coefficients are drawn from the set $\{+1, 0, -1\}$, which contrasts with the single-bit binary case where the coefficients exist in $\{+1, -1\}$. In return for the additional signal bit needed to describe its coefficients, the ternary filter will exhibit a higher Signal to Quantization Noise Ratio (SQNR) compared to the binary case (see Table 3.1, below).

This filter architecture comprises two parts: a ternary FIR filter stage followed by a re-modulator (i.e., IIR filter). The Ternary FIR filter (Figure 3.2) exhibits a conventional transversal structure and its output *y(k)* is in a multi-bit format. The IIR re-modulator filter follows the ternary FIR filter to transform its output back to single-bit format at the cost of an increase in chip area and lower performance. As will be shown below, regardless of the addition of the separate IIR re-modulator, and compared to the work reported in [2, 70], the hardware FPGA implementation of this filter at an equivalent filter order has superior performance at the cost of slightly more

hardware. Furthermore, additional coefficient quantization is unnecessary as coefficients are already in single-bit format.

3.3.1 Ternary FIR Filter (TFF)

Although the taps of the ternary filter are constrained to the set $\{+1,\ 0, -1\}$, it can be seen that its overall architecture is identical to the direct form of its multi-bit counterpart (Figure 3.2). The ternary filter output *y(k)* is given by the convolution of the taps h_i and the input signal *x(k)* as follows:

$$y(k) = \sum_{i=0}^{M} h_i x_{k-i} \tag{3.1}$$

where *M* is the order of the filter (≡ number of taps) and *h* represents the ternary FIR filter coefficients $\{+1, 0, -1\}$. The ternary format of the taps can be generated using a second order sigma delta modulator (ΣΔM) as reported in [10, 69]. The essential requirements for this ΣΔM structure (Figure 3.3) are to achieve a flat pass band across the desired frequency band and for the output of the quantizer to be in a ternary format. The z-domain transfer functions of the second order ΣΔM (Figure 3.3) is:

$$H(z) = N(z)z^{-1} + E(z)(1 - 2z^{-1} + z^{-2}) \tag{3.2}$$

where *N (z)* represents the target impulse response and *E(z)* is the quantization noise transfer function. In ΣΔM the inherent filtering term, $(1 - 2z^{-1} + z^{-2})$ is responsible for the noise shaping effect. The frequency response of this ΣΔM is given by:

$$H_{\Sigma\Delta T}(e^{j\Omega}) = N(e^{j\Omega})e^{-j\Omega} + E(e^{j\Omega})(1 - 2e^{-j\Omega} + e^{-2j\Omega}) \tag{3.3}$$

where $\Omega = 2\pi f / f_S$ is the normalized frequency (radians). In the same way, the in-band noise of a 2nd order sigma-delta modulator can be defined as:

$$\sigma_{ey}^2 = \sigma_e^2 \frac{\pi^4}{5}\left(\frac{2f_B}{f_s}\right)^5 \tag{3.4}$$

where $2f_B / fs = 1/OSR$, $\sigma_e^2 = \Delta^2/12$ is the mean square error, and the step size (Δ) has its standard definition. Considering N bits ADC with 2^N as number of quantization levels then $\Delta = (x_{\max} - x_{\min})/2^N$ [86]. The signal-to-quantization noise ratio (SQNR) of the 2nd order sigma-delta modulator can be defined as:

$$SQNR = 10\log(\sigma_x^2) - 10\log(\sigma_e^2) - 10\log\left(\frac{\pi^4}{5}\right) + 15.05r(dB) \tag{3.5}$$

where σ_x^2 and σ_e^2 are signal and quantization error power or variance and OSR is represented by $f_s / 2f_B = 2^r$. This function illustrates the direct relationship between the SQNR and resolution of ADC. Every single-bit increase in a 2nd order ΣΔM ADC will increase SQNR by 6-dB. Similarly, an increase of half a bit in ADC will add 3-dB of SQNR. On the other hand, every doubling of OSR will add approximately 15-dB SQNR [86].

3.3.2 Generation of Ternary FIR filter in MATLAB

The generation of a ternary FIR filter (e.g., in MATLAB) commences by selecting the Target Impulse Response (TIR). In the following sections we have used a low pass filter example with the following specifications: Sampling Frequency 8000Hz, Pass band 0-800Hz, Stop band 1200-4000Hz, Pass band Ripple (δp) 1.5dB and Stop band Attenuation (δs) of 90dB. The Target Impulse Response was generated using the Remez exchange algorithm. The optimum order of the filter for these specifications was found to be 63. The desired TIR with pass band and stop band attenuation values of 0.2π and 0.3π respectively and with 90-dB of stop band attenuation is shown in Figure 3.4 .

To satisfy the input oversampling requirement of the sigma-delta modulator, the coefficients must be scaled before encoding into the ternary format so that its peak input operates within the maximum signal-to-quantization-noise ratio (SQNR), fully utilizing the available dynamic range. FFT is one of the efficient scaling techniques reported in [58] and has been used here. The taps are encoded into a ternary format after scaling (i.e., oversampling). It is worth noting here that using a ternary format for the coefficients results in better SQNR compared to binary [87].

The Ternary filter (i.e., with ternary coefficients) exhibits the same impulse response as the TIR (specifically, in the pass band) but with a number of taps

proportional to the OSR [81]. The ternary coefficients, *r(k)*, represent the output of the sigma-delta modulator and were derived using:

$$r(k) = \begin{cases} +1 & w(k) > \dfrac{\beta}{4} \\ 0 & -\dfrac{\beta}{4} < w(k) < \dfrac{\beta}{4} \\ -1 & w(k) < -\dfrac{\beta}{4} \end{cases} \tag{3.6}$$

where $[-\beta/2, \beta/2]$ is the dynamic range of the sigma delta modulator. The ternary coefficients and the binary input stream generated at this stage according to the filter specification given above were used to simulate the FIR filter implemented in VHDL described below.

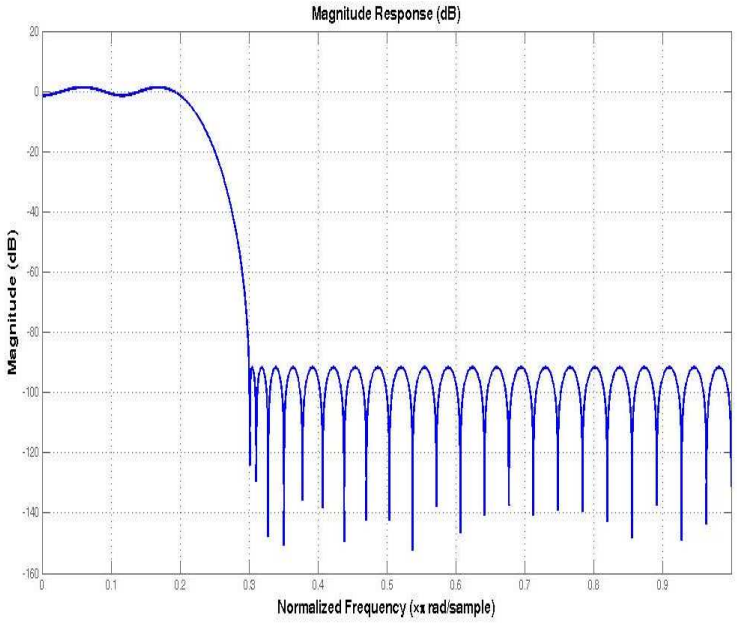


Figure 3.4 Target Impulse Response by Remez Exchange Algorithm

3.3.3 IIR Re-modulator

The output of the ternary FIR filter is in multi-bit format and includes a high frequency noise component. To overcome this an efficient re-modulator IIR filter was proposed in [10] (see Figure 3.1). The transfer function of the IIR sigma-delta re-modulator filters is:

$$H_{IIR}(z) = H_{IIRS}(z) + H_{IIRN}(z) \tag{3.7}$$

where S and N represents the signal and noise. H_{IIRS} is given by:

$$H_{IIRS}(z) = \frac{v.z^{-1}}{1-(1-u)z^{-1}} \tag{3.8}$$

and H_{IIRN} by:

$$H_{IIRN}(z) = \frac{(1-z^{-1})^3}{1-(1-u)z^{-1}}. \tag{3.9}$$

Finally, the overall frequency response of the filter as shown in Figure 3.1 can be described by:

$$H_{FIL}(e^{j\Omega}) = H_{\Sigma\Delta}(e^{-j\Omega}).H_{IIR}(e^{-j\Omega}) \tag{3.10}$$

From (3.7) and (3.11) we obtain:

$$H_{FIL}(e^{j\Omega}) = H_{\Sigma\Delta}(e^{-j\Omega}).(H_{IIRS}(e^{-j\Omega}) + H_{IIRN}(e^{-j\Omega}) \tag{3.11}$$

which can be further expressed as:

$$H_{FIL}(e^{j\Omega}) = \frac{N(e^{j\Omega})[e^{-j\Omega} + e^{-2j\Omega}(v-1)}{1-(1-u)e^{-j\Omega}} + \frac{E(e^{j\Omega})}{1-(1-u)e^{-j\Omega}} \begin{bmatrix} 1+e^{-j\Omega}(v-3) \\ +e^{-2j\Omega}(3-2v) \\ +e^{-3j\Omega}(v-1) \end{bmatrix} \quad (3.12)$$

As is evident from the general form of their transfer function, the overall simplicity of short word length techniques can result in very simple hardware implementation, especially on fine-grained devices such as FPGAs. An implementation of this ternary filter is discussed in next section.

3.4 FIR Filter Design in VHDL

This section discusses the overall architecture of the single-bit FIR filter implemented in VHDL.

3.4.1 Single-bit Ternary FIR-like Filter Hardware Implementation

The basic structure of a SBTFF (Figure 3.5) comprises two sections: multiplication of the coefficient taps with the binary input followed by the addition of the partial products. As the ΣΔM typically operates at a high Oversampling Ratio (OSR), this may result in a large number of taps: in the range of 2^N. Thus, for example, a multi-bit FIR filter with 64 taps will require 2048 ternary taps at an OSR of 32. The implementation of Figure 3.5 divides this into *N* coefficient multiply blocks followed

by an adder tree with $\log_2 N$ levels to perform the summation. As described earlier, many algorithms have been developed to reduce latency as well as improve the performance of multi-bit (i.e., conventional binary) FIR filters [2, 66-67]. To achieve improved performance with a smaller number of LUTs, we have focused on techniques that map efficiently onto FPGA organizations but that are also suitable for ASIC implementation.

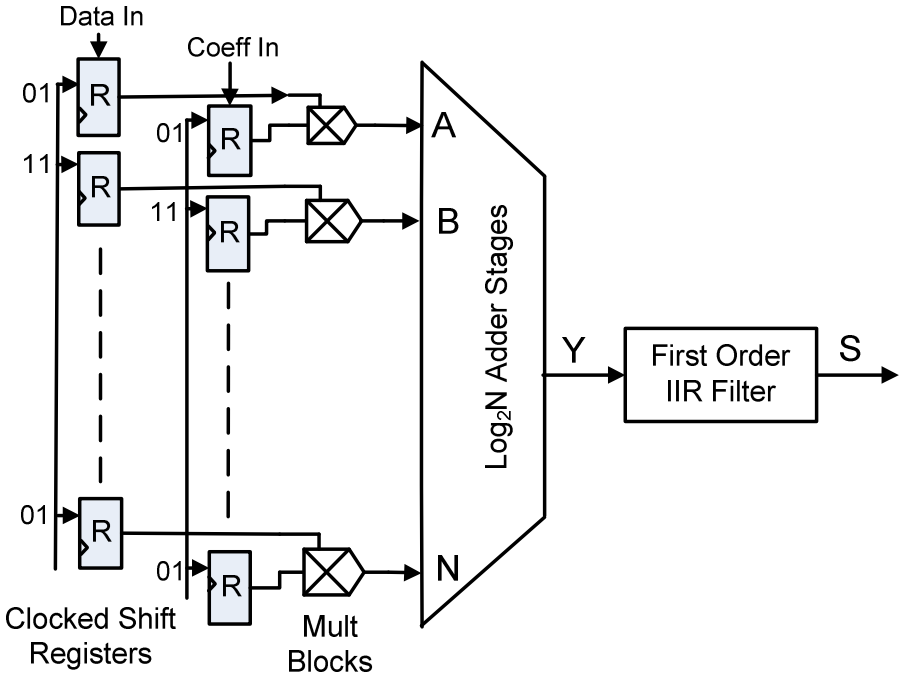


Figure 3.5 Block Diagram of SBTFF in Hardware

To deal with signed-bit arithmetic in FPGAs, 2's complement format is a reasonable choice. The coefficient symbols $\{+1, 0, -1\}$ can be easily mapped to two-bit numbers in 2's complement as: $+1 \rightarrow 01$, $0 \rightarrow 00$, $-1 \rightarrow 11$. Note that, while 2's complement can simplify the arithmetic, any other dual-rail (2-bit) format is equally

applicable here. Using a binary tree structure to sum the partial products over N = 2048 implies eleven stages and a final multi-bit result of $\pm N$, thus requiring a total of 13 bits to completely express the full output range of ±2048. As mentioned above, single-bit filters reduce complex multiplication structures such as those employed in [2, 70-71] to simple AND-OR logic functions that can typically be mapped to a single LUT.

3.4.2 Ternary Multiplier and Adder Modules

A small fragment of the adder tree is shown in Figure 3.6. The overall number of addition blocks will halve at each successive adder stage while their length increases by one-bit, culminating in the final multi-bit output. If we consider 2048 data/coefficient pairs, so that the adder tree comprises 11 levels, the first few adder stages will have inputs in the range of 2 to 3 bits, which can easily be mapped to a single LUT in a typical FPGA architecture while the remainder will comprise small ripple-carry blocks from four to twelve bits long. Note that it would be equally possible to use optimized IP blocks created specifically for this purpose. In this chapter, we have taken a more general approach, so that our implementation results might be considered to be worse-case.

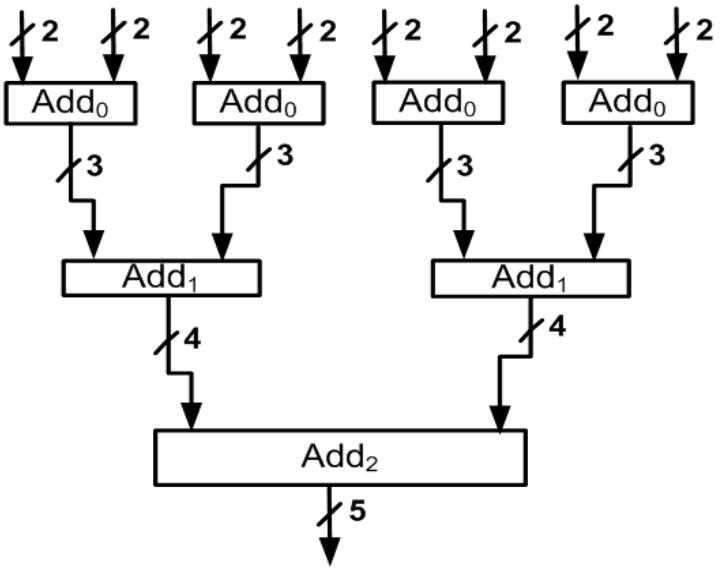


Figure 3.6 Two Level Fragment of the Adder Tree Structure.

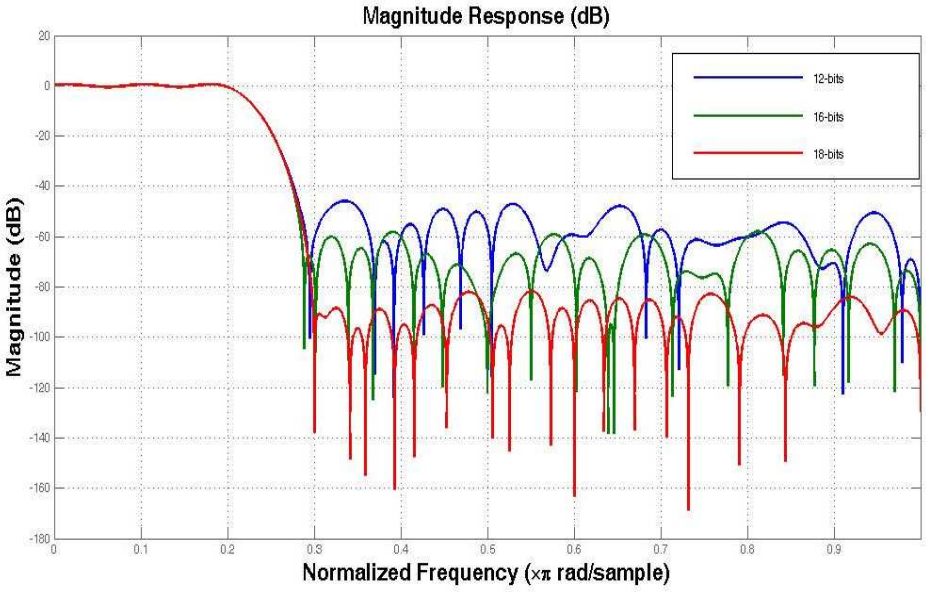


Figure 3.7 Frequency Response of the Target Filter at various coefficients bit-widths (=12, 16 and 18).

As highlighted above, the main advantage of the ternary design is its simple hardware implementation, especially with respect to the multiplier blocks that are typically the most complex modules in multi-bit filters. The Boolean logic of a ternary multiplier outputs (m_1, m_0) using ternary format for coefficients *(c_1c_0)* and inputs *(d_1d_0)* is given simply as:

$$m_1 = c_0.\overline{c_1}.d_0 + d_0.\overline{d_1}.c_1;\ \ m_0 = c_0 d_0. \tag{3.13}$$

where c_0 and c_1 are ternary coefficients and d_0, d_1 are binary data bits. This simple implementation of the single-bit multiplier and short adder modules results in a robust and efficient design that exhibits significant advantages over its complex multi-bit counterpart.

3.4.3 Multi-bit FIR Filter Design

As there is no essential architectural difference between single-bit and multi-bit filter organizations, (except for the additional IIR re-modulator part following the FIR filter in the single-bit case), similar design methods can be used for both. In the experiments reported here, the multi-bit filter coefficients were converted into fixed point using the fixed point toolbox available in MATLAB [88]. To maintain the fixed point format, double precision coefficients generated by MATLAB were initially converted into single-precision FP format. Coefficients were then further quantized with tight constraints into the required number of fixed point mantissa bits (i.e., 12, 16, 18 bits) [89]. As expected, some of the precision was lost after quantization so the

filter response shown in Figure 3.7 has greater ripple in the stop band compared with the target impulse response (Figure 3.4). It can be observed that, as expected, the stop-band ripple diminishes as number of mantissa bits increases. The filter response comes within about 1dB of the desired shape at a mantissa bit-width of 18.

3.4.4 Spectral Performance Comparison

Table 3.1 shows a comparison between single-bit and multi-bit filters on the basis of their theoretical spectral performance at a fixed filter order of 64 and with varying bit-widths. The binary (B) and ternary (T) SQNR has been included in this table simply to illustrate how it impacts upon the filter performance, using (3.5). However, using the ternary format results significant improvements in SQNR without any impact upon hardware area due to the 2's complement format mapping discussed in (section 3.4). In same way, the multi-bit filter order at both theoretical (6-dB) and practical (5-dB) SNR values are shown. Theoretical and practical values of multi-bit SNR are presented so the ternary SQNR can be correlated with its corresponding SNR.

Table 3.1 Signal-to-Noise Ratio Comparison of Single-bit and Multi-bit FIR Filter

N	Single-bit Filter			Multi-bit Filter		
	OSR	SQNR		Coefficient Bit Width	5dBpb	6dBpb
		Binary	Ternary			
64	8	34	43	8	40	48
	32	64	73	12	60	72
	64	79	88	16	80	96
	128	94	103	18	90	108

However, it should be noted that there are differences in the spectral performances of the filters shown in Table 3.1 that make them difficult to compare directly on a one-on-one basis. These experiments attempted to match the spectral performance of the two filter types as closely as possible although it is difficult to achieve both an OSR that is an exact power of 2 and at the same time match the relative spectral performance of the two types. In setting up the corresponding cases in Table 3.1, an attempt was made to reduce the difference between the spectral performances of the two filters without concern for their absolute performance relationship. For example, at an OSR of 8, the single-bit filter exhibits an SQNR of 43dB with ternary (T) and 34dB with binary (B) coefficients. This level of SQNR improvement obtained using SDM with ternary coefficients is consistent with the average gains of 9.0 dB and 7.0 dB reported in [53, 87] . The corresponding values for the nearest multi-bit filter with an approximately equivalent SNR (i.e., CBW = 8) are 40dB (5dB/bit) and 48dB (6dB/bit). Doubling the OSR (from 8 to 16) would raise the SQNR of the single-bit filter to 58dB with ternary format, moving it too far away from the corresponding 48dB value of the multi-bit filter to be comparable.

Coefficient bit width (CBW) is an important design issue for multi-bit filters. When synthesizing the filter hardware we chose the input bit-width to be the same as the coefficient width at each stage. In this way, a multi-bit filter with N=64 requires sixty four multiply blocks and six adder stages to achieve its final multi-bit output. Note that an alternative scheme could use just one multiplier operated sequentially (or,

indeed, any number between 1 and 64) with correspondingly lower overall area and performance. In this chapter, we are comparing two equivalent architectures that offer peak performance for a given technology.

3.5 Simulation Results and Discussion

Both filters were coded in VHDL and compiled, simulated and synthesized in Quartus-II 9.1, on Cyclone-III EP3C120F484C7 and Stratix-III EP3SL340F151C72 devices using a vector waveform file input. The filter coefficients and input bit-stream for single-bit and multi-bit filters, previously generated using MATLAB were held in block RAM. A simulation commenced by transferring the coefficients from memory to the input registers. Following this initialization stage, the data stream was serially shifted to the data registers. On each clock cycle, data and coefficients shift one bit, triggering new multiply-accumulate operations. The final (multi-bit) output stream was stored in another local memory bank. In the case of the single-bit FIR filter, the ternary filtered multi-bit output was further passed to the IIR filter for single-bit conversion.

3.5.1 Filter Area-Performance Analysis

Various area-performance tradeoffs for single-bit and multi-bit are presented for both non-pipelined (Table 3.2) and pipelined modes (Table 3.3). The appropriate number of ternary coefficients (TC) was obtained by multiplying the actual number of coefficients by the OSR. For example, at a fixed order of 64 and with an OSR of 32, the number of ternary coefficients (i.e., the order of the single-bit filter) would be

64 x 32 = 2048. On the other hand, the multi-bit simulations used an adder tree with an identical structure as the single-bit filters but using built-in multiplier macro blocks.

During the place and route steps, the maximum operating frequency (F_{MAX}) was constrained to a value somewhat higher than achievable for the given technology. The objective was to force the tools to generate a final routing that was comparable across devices. The approximate number of LUT elements was determined from the flow summary, while F_{MAX} has its usual definition as the maximum clock rate achievable with zero slack on the worse-case critical path(s). Although the implementations are directly comparable, the results may slightly vary in real applications as no account was taken of pin capacitance or specific design optimizations such as forcing the use of I/O registers.

It can be seen from Table 3.2 and Table 3.3 that the maximum performance of the single-bit filter is consistently superior to its corresponding multi-bit filter. For example, in non-pipelined mode the single-bit filter exhibits 40-50% higher maximum clock frequency (F_{MAX}) over the range of filter orders explored here. Even up to medium order filters (i.e., below TC=4096), the area cost of both approaches is the same to within 3-4%. It is not until the highest order simulated here (TC ≥ 8192) that the area of the single bit filter significantly exceeds that of its multi-bit counterpart. It is also worth nothing that, even the highest filter orders shown here still fit comfortably into the lowest cost FPGA device (the Cyclone-III).

Table 3.2 Area-Performance comparison of single-bit FIR vs. multi-bit filter: non-pipelined Mode.

Device	Single-bit			Multi-bit			
	Tern. Coeff	LUTs	F_{MAX}	N	W	LUTs	F_{MAX}
Cyclone-III	512	4089 (3%)	71.4	64	8	8860 (7%)	46.2
	2048	15603 (13%)	52.6		12	17045 (14%)	35.3
	4096	30894 (26%)	45.3		16	26838 (22%)	29.1
	8192	62747 (53%)	40.3		18	32547 (27%)	26.5
Stratix-III	512	3925 (1%)	129.8	64	8	5219 (2%)	86.5
	2048	14368 (5%)	97.3		12	10942 (5%)	69.1
	4096	28499 (11%)	82.8		16	17731 (7%)	57.5
	8192	55927 (21%)	69.6		18	21568 (8%)	51.2

Table 3.3 Area-Performance Comparison of Single-bit FIR vs. multi-bit Filter: pipelined Mode

Device	Single-bit			Multi-bit			
	Tern. Coeff	LUTs	F_{MAX}	N	W	LUTs	F_{MAX}
Cyclone-III	512	3963 (3%)	125.6	64	8	9020 (8%)	94.5
	2048	15399 (13%)	122		12	17079 (14%)	67.1
	4096	30607 (26%)	120		16	26890 (23%)	53.3
	8192	61029 (51%)	118		18	32586 (27%)	47.4
Stratix-III	512	3719 (1%)	240	64	8	4923 (1%)	258.3
	2048	14453 (5%)	237		12	10353 (4%)	199.0
	4096	28745 (11%)	237		16	16916 (7%)	158.8
	8192	57362 (21%)	231		18	20662 (8%)	139.7

To achieve a valid comparison with the area and performance simulation results given in [2] (that were carried out with fixed 8-bit coefficients), the single-bit ternary filter simulation was designed to achieve an equivalent spectral performance with 64 fixed coefficients. The equivalent area-performance simulation results of ternary filter obtained (at TC = 512, i.e. OSR = 8) has almost double the performance at comparable chip area , as reported in [2].

In pipelined mode, additional registers were placed between the adder stages, increasing the throughput of the filter at the cost of a moderate increase in the number of registers and a small increase in latency. Unlike the multi-bit filter single-bit filter greatly benefits from pipelining due to the simplicity of the multiplier (see Table 3.3). The single-bit filter organization achieved a maximum of 42% improvement in maximum operating frequency (F_{MAX}) over its corresponding multi-bit filter.

It can be seen in Table 3.3 that the performance of the multi-bit filter decreases linearly from 199MHz as the coefficient width increases (increasing the SNR). It declines by around 31% to 139MHz at 18-bits. In contrast, F_{MAX} is almost unchanged in the single-bit case under the same conditions (Table 3.3). The maximum operating frequency of 239MHz at 2048 coefficients reduces by only about 4% as the number of ternary coefficients is successively doubled to 8192.

However, it is also clear from Table 3.2 and Table 3.3 that the difference between the two approaches diminishes at lower OSR values, particularly in pipelined mode.

Predictably, the IIR demodulator circuit forming the final stage of the ternary FIR filter impacts its overall performance and becomes the limiting factor at small OSR values. Thus, the pipelined filter performance at an OSR of 8 remains almost same as at 32 or 64. On the other hand, at an OSR of 8 the filter performance continues to improve in non-pipelined mode (Table 3.2), becoming about 27% better at OSR of 32 and 38% at OSR of 64 respectively.

As the multiplier is the primary critical path in the multi-bit FIR filter structure, it is virtually impossible to optimally balance its pipeline. This might conceivably be achieved by modifying the internal stages of the multiplier but this was not possible in our work as we were already using highly optimized IP blocks in Quartus II. A disadvantage of these macro-blocks is their inflexibility. It was not possible to add internal pipelining to optimally balance the processing stages.

3.5.2 Filter Power Analysis

The Power Dissipation analysis of both the filters was performed in Quartus-II 9.1 using the "Power Play" Power Analysis Tool [90] after the generation of a signal activity file (.saf). The total power of a FPGA device is made up of I/O power, core static power and core dynamic power [90]. core static power was observed to be more or less constant across all designs so was not measured separately. The main impact on static power at the system level comes from the assignment of unused configurable logic blocks (CLBs) and routing inputs.

In general terms, dynamic power depends upon many factors including switching activity, design style, number of logic blocks and interconnects and input-output data bandwidth [2, 90]. It varies with frequency according to:

$$P = a.F.C.V^2 \qquad (3.14)$$

where a is the activity factor (broadly, the probability that a particular node will perform a transition at a given time), C is total load capacitance, F is the transition frequency (usually assumed to be equal to or directly proportional to the clock frequency) and V is supply voltage. Note that there is also a contribution from short-circuit current during switching, but it tends to be small when the input and output rise times are roughly equivalent [91] and is not reported separately in Power Play.

The area-performance results obtained in Table 3.2 and Table 3.3 identify the maximum operating frequency of the corresponding filter types achievable at specific filter orders and data lengths, using currently available FPGA technology. Equation (3.14) implies that at a given technology (i.e., fixed CV^2), the dynamic power will depend directly on both clock and node activity (aF). Thus, clock frequency is an important parameter when comparing these filters styles and, in general terms, two choices are possible:

to run each filter at its individual F_{MAX}. The resulting spectral performance will be quite different for the filters, making direct comparison difficult;

to operate the filters at a pair of related clock frequencies that results in equivalent spectral performance. In this case, the single-bit filter operating frequency would be OSR*F_S, where F_S is the frequency of the corresponding multi-bit filter. This is the method used in the the following analysis.

Table 3.4 Clock Frequency for Ternary and Multi-bit Filters pipelined and non-pipelined modes.

F_{8K} Process			F_{MAX} Process			
			Non-Pipelined		Pipelined	
Device	TClk (KHz)	MClk (KHz)	TClk (MHz)	MClk (MHz)	TClk (MHz)	MClk (MHz)
Cyclone -III	64	8	71.4	46.2	125.5	94.7
	256	8	52.6	35.3	122	67.1
	512	8	45.3	29.1	120	53.3
	1024	8	40.3	26.5	118	47.4
Stratix – III	64	8	129.8	86.7	240	258.3
	256	8	97.3	69.1	239	199.0
	512	8	82.8	57.5	237	158.8
	1024	8	69.6	51.2	231	139.7

TClk: Ternary Clock, MClk: Multi-bit Clock

The dynamic power simulations outlined below were conducted in two stages. Firstly, both the filters were simulated at their maximum clock frequency determined by the worse case F_{MAX} (from Table 3.2 and Table 3.3) for either the single-bit or the multi-bit filter, related via the performance of the filter (labelled F_{MAX} in the results tables).

In a second step, the two filter types were set up to achieve the specifications outlined in section 3.3.2, (i.e., at F_S = 8000Hz). As identified in Table 3.1, a single-bit filter can achieve an equivalent spectral performance to the multi-bit case by

increasing its OSR, so in this case the single-bit filter clock was obtained by multiplying the OSR to the Nyquist frequency (i.e., F_S*OSR). The multi-bit filter clock was kept at its Nyquist rate (8000Hz) throughout (labeled F_{8K} in the results tables).

Table 3.4 summarizes the various clock frequencies used for the simulated filter implementations in Cyclone III and Stratix III devices. In the F_{8K} case, both the filters clock remain the same in pipelined as well as non-pipelined modes because of clock is dependent upon F_S. In the F_{MAX} processes, the clock frequency varies according to the values obtained for the two modes given in Table 3.2 and Table 3.3.

The simulated dynamic power results are presented in Table 3.5 and Table 3.6 (see pp: 78 and 79). As expected, the high operating frequencies of the single-bit filters results in the majority of the power dissipated arising from the operation of the registers and their associated clock tree, starting at around 50% for the low clock-rate Cyclone-based filters and rising to greater than 96% for the high performance Stratix implementations. On the other hand, multi-bit filters dissipate most power in their combinational circuits with little (<10%) resulting from the register and clock operation. In both filters, I/O power is relatively constant or grows linearly as the order of the filter increases.

It can be seen that despite their much higher clock rates, the single-bit filters typically dissipate a small fraction of the power of their corresponding multi-bit filters,

e.g. around 20% for non-pipelined mode in Table 3.5. However, in pipelined mode, the effect of the larger number of registers (plus their control blocks) means that the single-bit filter power may exceed that of the multi-bit case (by around 10 to 20%) at the maximum clock rates for these filters.

On the other hand, the low clock frequencies for the F_{8K} process results in very small absolute power dissipation by both filters types with the I/O power dominating in all cases. Even so, the multi-bit filters can be seen (Table 3.6) to consume between 1.5 and 3 times the power of their equivalent single-bit filters. Of the range of filter configurations studied in this work, only in the case of largest filter (8192 coefficients), using the most aggressive technology (Stratix III) in fully pipelined mode (i.e., with the greatest number of registers) did the single-bit filter power exceed that of its corresponding multi-bit case—by about 30% in that case.

It is also notable that the single-bit filters are capable of a higher maximum bandwidth than their multi-bit counterparts, with correspondingly lower power dissipation. This offers an additional level of flexibility: it is possible to trade off power, area and performance over a wider range of filter spectral characteristics than is the case in multi-bit filters.

Table 3.5 Dynamic Power Dissipation: F_{MAX} Process.

Device	TClk (Mhz)	MClk (MHz)	#TC	#W	Dynamic Power Dissipation: Non-Pipelined Mode									
					Ternary (mW)					Multi-bit (mW)				
					CC	CCB	Reg	I/O	Total	CC	CCB	Reg	I/O	Total
C-III	71.4	46.2	2^8	8	8	21	17.3	15	61	292	9	12	21	335
	52.6	35.3	2^{11}	12	7	29	45	13	94	366	10	13	24	414
	45.3	29.1	2^{12}	16	7	39	80	12	138	603	12	16	27	658
	40.3	26.5	2^{13}	18	6	47	140	12	206	772	12	18	29	831
S-III	129.8	86.7	2^8	8	56	46	29	26	156	629	26	37	63	757
	97.3	69.1	2^{11}	12	44	65	72	22	203	831	28	36	53	947
	82.8	57.5	2^{12}	16	34	81	115	19	248	1704	38	58	66	1865
	69.6	51.2	2^{13}	18	27	114	189	17	347	1755	34	58	60	1907
Dynamic Power Dissipation: Pipelined Mode														
C-III	125.5	95	2^8	8	5	31	43	22	101	172	16	68	38	294
	122	67	2^{11}	12	6	51	151	22	229	343	56	80	41	519
	120	53	2^{12}	16	6	85	297	22	411	374	25	85	40	522
	118	47	2^{13}	18	6	124	592	23	745	519	25	92	42	677
S-III	240	258	2^8	8	28	76	113	47	264	284	46	187	103	610
	239	199	2^{11}	12	29	157	374	46	606	478	70	249	121	919
	237	159	2^{12}	16	30	235	721	47	1033	907	69	297	124	1397
	231	140	2^{13}	18	30	430	1409	48	1917	1076	68	308	123	1575

#TC: Number of Ternary Coefficients, #W: Multi-bit filter bit width, CC: Combinational Circuits, CCB: Clock Control Block, Reg: Registers, Total: Sum of dynamic power dissipation of CC, CCB, Reg and I/O

Table 3.6 Dynamic Power Dissipation: F_{8K} Process

Device	TClk (KHz)	MClk (KHz)	#TC	#W	Dynamic Power Dissipation: Non-Pipelined Mode									
					Ternary (mW)[1]					Multi-bit (mW)				
					CC	CCB	Reg	I/O	Total	CC	CCB	Reg	I/O	Total
C-III	64	8	512	8	0	0	0	3.0	3.0	0.05	0	0	5.91	6.0
	256	8	2048	12	0	0	0	3.4	3.4	0.09	0	0	7.41	7.5
	512	8	4096	16	0	0	0	3.6	3.6	0.17	0	0	8.91	9.1
	1024	8	8192	18	0.01	0.6	0.92	3.7	5.25	0.24	0	0	9.66	9.9
S-III	64	8	512	8	0	0	0	0.8	0.85	0.04	0	0	1.77	1.8
	256	8	2048	12	0	0	0	1.0	1.0	0.08	0	0	2.26	2.3
	512	8	4096	16	0	0	0	1.0	1.0	0.16	0	0.01	2.73	2.9
	1024	8	8192	18	0	1	0.94	1.1	3.0	0.21	0	0.01	2.97	3.2
Dynamic Power Dissipation: Pipelined Mode														
C-III	64	8	512	8	0	0	0	3.0	3.0	0.02	0	0.01	5.91	6.0
	256	8	2048	12	0	0	0	3.3	3.4	0.03	0	0.01	7.41	7.5
	512	8	4096	16	0	0	0	3.7	3.8	0.06	0	0.01	8.91	9.0
	1024	8	8192	18	0.01	0.52	1.64	4.2	6.4	0.09	0	0.02	9.66	9.8
S-III	64	8	512	8	0	0	0	0.8	0.8	0.01	0	0.01	1.77	1.8
	256	8	2048	12	0	0	0	1.0	1.0	0.02	0	0.01	2.26	2.3
	512	8	4096	16	0	0	0	1.04	1.04	0.06	0	0.02	2.73	2.8
	1024	8	8192	18	0	0.84	1.77	1.95	4.6	0.06	0	0.02	2.97	3.1

#TC: Number of Ternary Coefficients, #W: Multi-bit filter bit width, CC: Combinational Circuits, CCB: Clock Control Block, Reg: Registers. Total: Sum of dynamic power dissipation of CC, CCB, Reg and I/O, [1] the low (8K) clock rate in this case results in power dissipation levels in the nW range, recorded above as zero. The power components might not exactly sum to the total power due to rounding.

3.6 Stability Analysis of ΣΔM Based Single-bit IIR Filter

The work presented earlier in this chapter, we have shown that Short Word Length (often called single-bit) or ternary) DSP systems offer better area-performance-power tradeoffs compared to their multi-bit counterparts. This performance can further be improved at equivalent level by increasing the order of the sigma-delta modulators or increasing the oversampling ratio (OSR). Increasing the order of the SDM increases the signal-to-noise ratio at lower OSRs, potentially reducing the system complexity at

the cost of a higher probability of instability [33]. The major cause of instable nature of SDM is due to its non-linear behaviour. The main sources of nonlinearity in sigma-delta modulators are the 1-bit quantizer, operational amplifier slew rate and nonlinear DC gain, and nonlinear switching response [92]. The stability of a SDM can be assured with proper statistical characteristics of its input, the quantizer gain, and its order.

Various approaches have been taken to model the quantizer stage in order to determine the theoretical stability of the SDM, especially at higher orders. Linearizing the system by replacing the quantizer with a variable gain [34] does allow arbitrary inputs and initial conditions to be applied and provides insight into its stability but does not result in an accurate solution. An alternative approach that models the nonlinear behaviour of quantizer by setting two different gain parameters one for signal and another for noise was proposed by Ardalan and Palous [35], and further studied by Lota and Al-Janabi [92]. In all these techniques, a major parameter that assures the stability of sigma delta modulator is the quantizer input, which ultimately depends upon the sigma delta modulator input. If the input to the quantizer remains within defined limits then stability can be assured [31].

In this connection, we consider the nonlinear behaviour of the 2^{nd} order sigma delta modulator that forms part of a single-bit ternary FIR-like filter (Figure 3.1). In general, second order sigma delta modulators are likely to be stable and are less prone to the limit cycle, chaos and idle tone if the input and quantizer gain bounds are within

certain limits. It was shown in [54] that the system will remain stable if the feedback loop parameter and quantizer gain remain below 1.3. However, this claim is not general and will depend on the topology, input type and other factors.

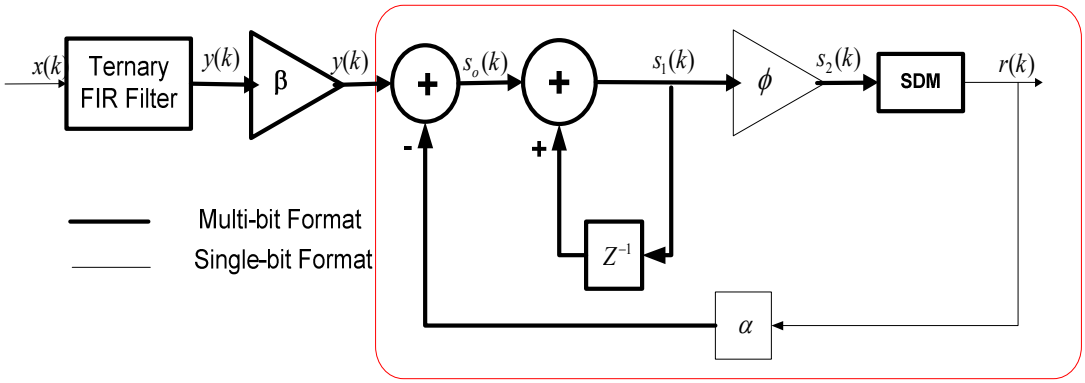


Figure 3.8 Proposed single-bit ternary filter with a gain factor inside the loop

We analyse the stability of the ΣΔM component using a linearized method combined with a root locus approach. Thus, the more rigorous description function approach of [35] is not warranted in this case and the linear method gives sufficient insight into the stability of the system and supports an adequate analysis of its performance [31]. Unlike previous work, here the second order ΣΔM stability bounds that are widely accepted and reported in the literature [93] are included. By taking these bounds into account, we propose a new design that ensures the stability of the single-bit ternary FIR filter by allowing greater control over both the input to the ΣΔM and the quantizer gain. The overall transfer function of an example single-bit ternary

FIR filter has been derived and based on a linear analysis of the system in MATLAB; we present new bounds for the IIR loop parameters that ensure the stability of the filter.

3.7 Stability of Single-bit Ternary FIR-like filter

As FIR filters are inherently non-recursive and stable, the overall stability of the filter in (Figure 3.1) depends only on the IIR re-modulator stage. In general terms, it is relatively easy to achieve stable operation in 2nd order ΣΔM blocks (Figure 3.3), by maintaining tight limits on their input, state parameters and quantizer gain [33, 93]. In [93], second order modulators (Figure 3.3) stability bounds are reported with DC inputs if all the input samples remain in the range $s_2(k) < 1$. Difference equations for the second order ΣΔM shown in Figure 3.3 are given below:

$$r(k) = \operatorname{sgn}\left[g_2(k)\right] \tag{3.15}$$

$$\begin{bmatrix} g_1(k+1) \\ g_2(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} g_1(k) \\ g_2(k) \end{bmatrix} + \begin{bmatrix} -1 \\ -2 \end{bmatrix} r(k) + \begin{bmatrix} 1 \\ 1 \end{bmatrix} s_2 \tag{3.16}$$

The following upper bounds are reported in [93] for internal state variables:

$$|g_1| \le |s_2| + 2 \text{ and } |g_2| \le \frac{(5 - |s_2|)^2}{8(1 - |s_2|)} \tag{3.17}$$

where g_1 and g_2 are the loop filter outputs (see Figure 3.3) and s_2 is the input to the sigma-delta modulator. In [93] these bounds are reported for a DC input but may become tighter for any other input type: sinusoidal for example.

In [31], it is shown that even $|s_2(k)| < 0.3$ can require large internal state variables. Therefore it is wise to apply the limit $|s_2(k)| < 1$ in order to avoid large states at the second integrator [31]. Using a more robust non-linear approach, similar simulation based bounds are reported in [92] for a second order modulator with DC and sinusoidal inputs. However, the presence of the ΣΔM inside the IIR loop in Figure 3.8greatly complicates the analysis of its overall stability and makes predicting SNR more difficult [39, 54]. The main advantage of employing higher order ΣΔM blocks will be better SNR and reduced the chip area, which needs to be weighed against the impact of greatly increased complexity.

3.8 Proposed Design of SBTFF

In the original filter topology (Figure 3.1), two gain parameters *u* and *v* were used to control its overall stability [54]. The IIR filter was first simulated in MATLAB assuming a sinusoidal input with $f_o = 2000Hz$, OSR=128, amplitude i.e., $A = 0.5$ and that the 2nd order ΣΔM was unconditionally stable. Simulation of the IIR filter with *u* = 0.1 and *v* = 0.001 produces a linearly increasing input to the ΣΔM that leads to the quantizer input(i.e., $|g_2(k)|$) increasing in an unbounded manner according to the state-space model upper bounds given by (2) as shown in Figure 3.9.

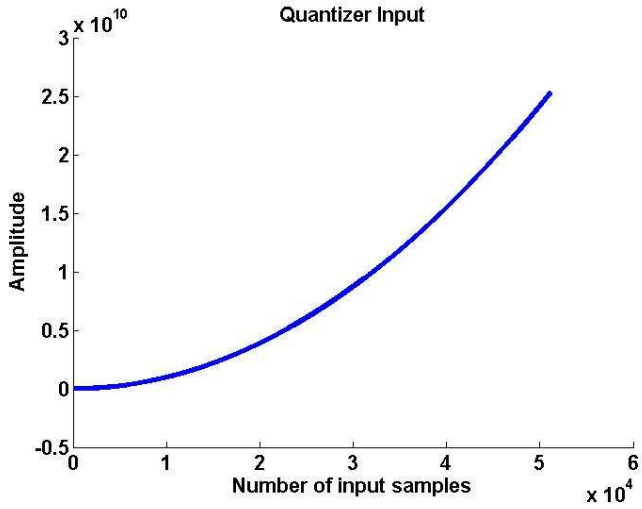


Figure 3.9 ΣΔM quantizer input $g_2(k)$ at sinusoidal excitation

In the examples that follow below, the FIR filter convolution output is in the range of ± 1024, which a value of $v = 0.001$ will scale to ±1 at the input of the IIR filter. Larger values of v will increase the overall filter gain, but will result in larger peak values at the IIR input, which will ultimately cause the ΣΔM stage to become unstable, particularly under low frequency and DC signal conditions.

To improve the control of the IIR loop we have introduced a new gain (ϕ) parameter inside the loop as shown in Figure 3.8. Its major advantage is that input to the sigma delta modulator can be easily controlled based on its dynamic range. The original filter gain ($v \leq 1$) reduces the input range and therefore its bit-width. In this paper, we have set $v = 1$. The adjustment of ϕ depends upon the maximum convolution output from the ternary FIR filter. With a typical speech filter simulated in MATLAB[82], the optimum ϕ that resulted in the input to the sigma delta modulator

constrained to the range of ± 0.5 was found to equivalent to *OSR* $\times$ *8*. These simulations were run 10^5 times to ensure the stability of the system and found the quantizer input that fluctuates in the range $|g_2(k)| \leq 2.75$ that follows the bounds as given in (3.15) and (3.16). Hence, we can consider that the overall system remains within its stable region.

The overall transfer function of the single-bit ternary FIR filter together with the proposed IIR re-modulator will remain same as given in [10] i.e.:

$$H(z) = N(z)z^{-1} + E(z)(1 - 2z^{-1} + z^{-2}) \tag{3.18}$$

where *N (z)* represents the target impulse response and *E(z)* is the quantization noise transfer function. With the new parameter ϕ inside the IIR loop, the signal transfer function and noise transfer functions of the IIR filter become:

$$H_{IIR}(z) = H_{IIRS}(z) + H_{IIRN}(z) \tag{3.19}$$

$$H_{IIRS}(z) = \frac{vz^{-1}}{1 - (1 - u\phi)z^{-1}} \tag{3.20}$$

$$H_{IIRS}(z) = \frac{(1 - z^{-1})^3}{1 - (1 - u\phi)z^{-1}} \tag{3.21}$$

Finally, the overall frequency response of the filter as shown in Figure 3.8 can be described as:

$$H_{FIL}(e^{j\Omega}) = H_{\Sigma\Delta}(e^{-j\Omega}).H_{IIR}(e^{-j\Omega}) \tag{3.22}$$

From (3.18) and (3.19) we obtain:

$$H_{FIL}(e^{j\Omega}) = H_{\Sigma\Delta}(e^{-j\Omega}).(H_{IIRS}(e^{-j\Omega}) + H_{IIRN}(e^{-j\Omega}) \tag{3.23}$$

that can be further expressed as:

$$H_{ov}(e^{j\Omega}) = \frac{N(e^{j\Omega})[e^{-j\Omega} + e^{-2j\Omega}(v-1)}{1-(1-u\phi)e^{-j\Omega}} + \frac{E(e^{j\Omega})}{1-(1-u\phi)e^{-j\Omega}} \begin{bmatrix} 1 + e^{-j\Omega}(v-3) \\ + e^{-2j\Omega}(3-2v) \\ + e^{-3j\Omega}(v-1) \end{bmatrix} \tag{3.24}$$

3.9 Stability Analysis by Root Locus Technique

Due to non-linear behaviour of sigma delta modulators, its stability analysis is difficult and even a complex mathematical approach may have minimal success [33]. However, a linearized model combined with a root locus approach offers a way to attain insight into the stability of the system with reasonable accuracy. Using this approach, the nonlinear quantizer component is replaced by a linear model called quantizer "gain" i.e., γ, that can be varied to change the closed loop poles of the system so that the stability can be analysed. The quantizer gain is defined as the amplitude (voltage) ratio of quantizer output to its input. With a linear quantizer model, the second order χ is redrawn as shown in Figure 3.10. Using this linear model of the

ΣΔM within the IIR re-modulator filter, the signal and noise transfer functions become:

$$H_{IIRS} = \frac{\gamma z^{-1}}{1 + (2\gamma - 3 + u.\phi.\gamma)z^{-1} + (3 - 3\gamma)z^{-2} + (\gamma - 1)z^{-3}} \tag{3.25}$$

$$H_{IIRN} = \frac{(1 - z^{-1})^3}{1 + (2\gamma - 3 + u.\phi.\gamma)z^{-1} + (3 - 3\gamma)z^{-2} + (\gamma - 1)z^{-3}} \tag{3.26}$$

The stability of the system is dependent upon the poles of the closed loop transfer function that can be positioned by varying the quantizer gain parameter (γ) and the feedback loop gain parameter, *u* and ϕ. The system will be stable if all of the poles remain inside the unit circle.

3.10 Simulation Results and Discussion

The single-bit IIR filter with a linearized model of the quantizer was simulated in MATLAB using the SBTFF model proposed in [10]. Simulations were carried out by varying the quantizer gain (γ) and the feedback loop gain (*u*). The intention was to determine the maximum range over which the system remained stable. The simulated results with varying parameters are given in Table I with and without ϕ. It can be seen that IIR filter is stable for $\gamma \leq 1.32$.

Further investigation shows that in the case where the quantizer gain increases beyond 1.3, the stability of the system sets a maximum limit of the feedback parameter

u at 0.16. Hence, there is a direct tradeoff between these two gain parameters. This limit is of great interest because it is very difficult to control the quantizer gain (especially at higher filter orders), due to its non-linear characteristics.

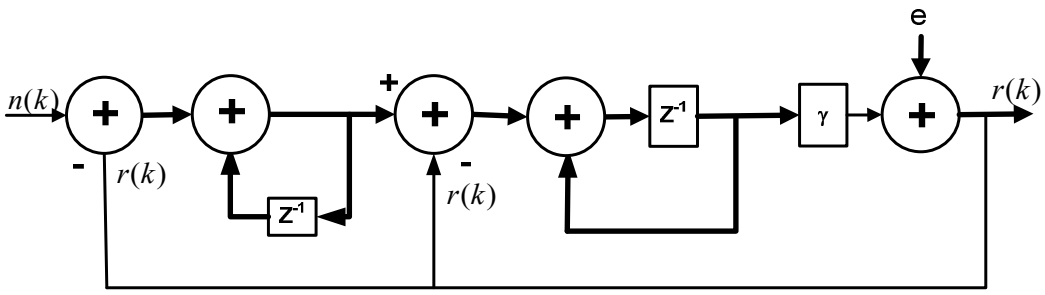
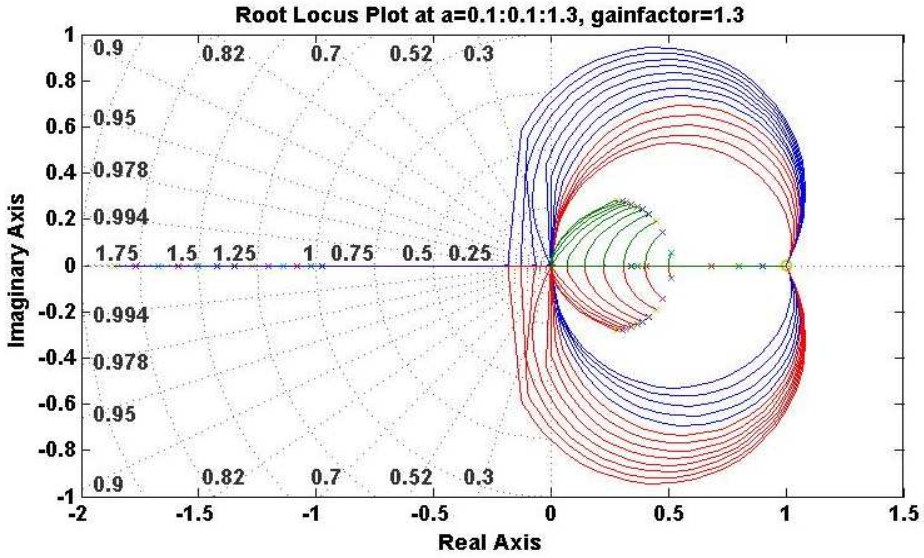


Figure 3.10 Linear Model of the 2nd order ΣΔM.

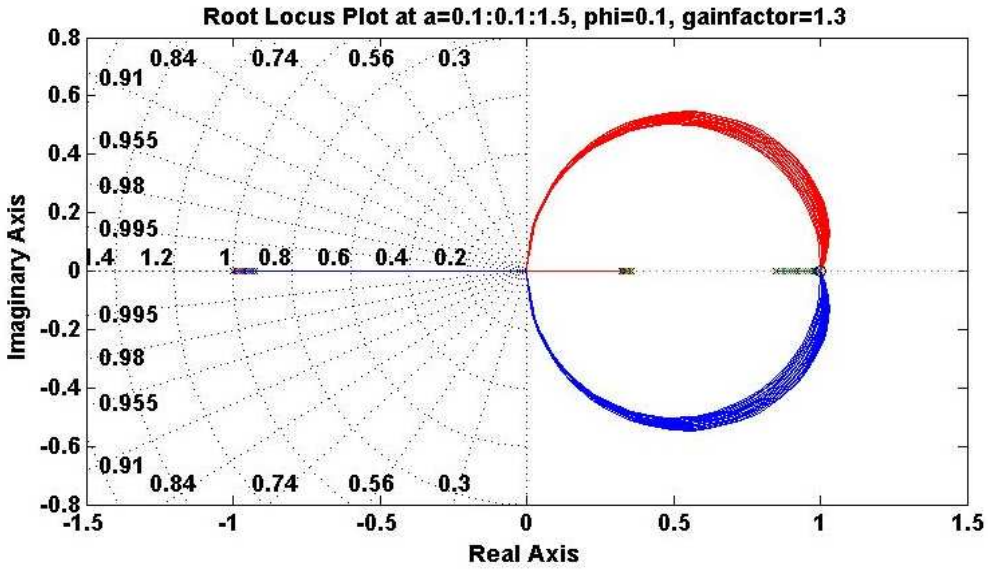
The simulations were undertaken with a fixed value of $\phi = 0.1$ and different combinations of quantizer and feedback gain. The simulation results show that the upper bound of the feedback loop parameters has increased to 1.5, while the quantizer gain limit still remains at 1.3. Moreover, adding a new parameter inside the loop restricts the IIR loop gain, and hence it can be utilized to control the system stability and the quantizer input gain. Additionally, fewer data bits may be required to achieve an equivalent SNR, reducing hardware costs and improving the performance of the system especially when mapping the hardware to a Field Programmable Gate Array (FPGA). This approach may result new area-performance tradeoffs between single-bit systems and multi-bit systems in hardware.

Table 3.7 : IIR loop stability analysis using root locus with varying quantizer gain (γ) and feedback loop gain (α) parameters with and without proposed design

γ	α	β	Closed loop poles	
			Real part (no ϕ)	Real Part (with ϕ)
0.1	0.1	0.1	0.94080675865268	0.99000915736323
			0.94080675865268	0.90449542131838
			0.90838648269463	0.90449542131838
1.3	1.3		-1.86043316132296	-0.9877103273592
			0.28521658066148	0.86931850553941
			0.28521658066148	0.34939182181984
0.1	1.3		1.01155389323414	0.95016736756146
			1.01155389323414	0.95016736756146
			0.64689221353171	0.88666526487706
1.3	0.1		-0.97256004173084	-0.9291403479063
			0.89971245124613	0.98999976452847
			0.34284759048471	0.32614058337786
1.3	1.5		-2.058117027368017	-0.9980028784224
			0.254058513684008	0.84889514559254
			0.254058513684008	0.35410773282989



(a) $\phi = 1$; $\gamma = 1.3$; $0.1 \leq \alpha \leq 1.3$



(b) $\phi = 0.1$; $\gamma = 1.3$; $0.1 \leq \alpha \leq 1.5$

Figure 3.11 Root Locus Plots with and without gain factor (ϕ)

The root locus plots (Figure 3.11) were generated in MATLAB with and without ϕ (i.e., with ϕ set to 1.0) at the fixed upper bound of $\gamma = 1.3$ and with the loop gain parameter α varied over the range 0.1 to 1.5. Figure 3.11 (a), for which $\phi=1$, shows that the original system becomes unstable for any value of $\alpha \geq 0.16$ when the quantizer gain is at its maximum value. By contrast, Figure 3.11(b) shows that reducing the value of ϕ (to 0.1 in this example) ensures that the system remains stable for all α up to 1.5.

3.11 Summary

In this chapter, single-bit FIR-like filter is investigated for area-performance-power tradeoffs implicit in VHDL implementations of single-bit and multi-bit FIR filters. In

general terms, it was found that using single-bit techniques in a FPGA environment results in superior performance at a cost of slightly more area at higher filter orders. This is largely due to a characteristic of typical FPGA architectures. As the primary building block ("logic element") in these devices comprises a small partitioned LUT plus one or more flip-flops (FFs), where one part of the logic element is used, the other is still available to be used in another part of the circuit. The overall area metric is therefore determined by the greater of the LUT and FF counts. It was found that a clock frequency of 250MHz is easily achievable in a high performance FPGA device. This would, for example, readily handle a 4MHz video stream at OSR of 64 in pipelined mode using about 5% of the available area of a mid-range commercial FPGA device.

The dynamic power dissipation figures of both the filter types were compared with equal clock rates as well as at the highest clock rates for which equivalent performance could be established. In all cases, the maximum performance of the multi-bit filters was the limiting constraint. It was found that at almost all clock rates, single-bit filters dissipate significantly less power than their equivalent multi-bit filters. The largest filter studied in this work represents the only case where the single-bit filter power exceeds that of its corresponding multi-bit case, and then only using the highest performance technology in fully pipelined mode.

In the continuation of single-bit FIR-like FPGA analysis, we have investigated the stability of the single-bit IIR filter by using a linearized model and a root locus

approach. In this approach, the ΣΔM quantizer is replaced by a linear quantizer gain. Previous simulations found that the system remains stable only if the quantizer gain remains less than 1.3 and the feedback parameter is kept within a lower bound of 0.16. Otherwise, the system becomes unstable immediately because of the positive feedback in the IIR loop.

To overcome this issue, a new design is proposed that regulates the input to the ΣΔM and ensures the stability of the overall system. With this new design, the IIR loop feedback parameter has been relaxed and can increase to 1.5 without compromising the stability of the system. The gain function adds little to the complexity of the filter, ensuring it can be efficiently mapped to hardware with fewer data bits, saving chip area and improving the system performance.

In the next chapter we will investigate the effects that different encoding techniques have on the Ternary FIR Filter in hardware implementation and on overall Single-bit FIR-like filter as well.

Chapter – 4

FPGA Analysis of Sigma-Delta Modulated Ternary FIR Filter with Alternative Encoding Techniques

4.1 Introduction

In chapter 3, FPGA design and analysis of SBTFF was presented in comparison to its counterpart multi-bit system. It is shown that SBTFF filter offers better area-performance-power tradeoffs compared to its counterpart multi-bit filter. The two filters were designed using simplest 2's complement encoding technique. However, there are number of encoding techniques that are frequently used in multi-bit system to minimize multiplier complexity and improve the area-performance tradeoffs [94].

In this chapter, we have chosen three classic encoding techniques i.e., 2's complement, redundant binary signed digit (RBSD) and canonical signed digit (CSD) to investigate the effect of the symbol encoding on the area and performance of short word length FIR filters. Through simulation, the area and performance of an example

filter are analysed in both fully pipelined and non-pipelined modes for all three techniques. It will be shown that, in contrast to their multi-bit counterparts, in the case of these short word-length filters the simplest encoding offers the best performance and area characteristics. This investigation is based on my published work [52, 69, 81, 83].

4.2 The Ternary FIR Filter (TFF)

As already discussed in chapter 3 the TFF (see Figure 4.1) output *y(k)* is given by the convolution of the ternary taps h_i and the input signal $x(k)$ as follows:

$$y(k) = \sum_{i=0}^{M} h_i x_{k-i} \tag{4.1}$$

where M is the order of the filter (≡ number of taps). The taps are generated by using the ΣΔM of the target impulse response. As the frequency response of the filter is directly related to the over-sampling ratio (OSR) and the number of taps, these filters typically require a very large number of coefficient taps (M). Thus, it is the parallel addition of the partial products that forms the processing "bottleneck" in this architecture. This issue will be addressed in the following section.

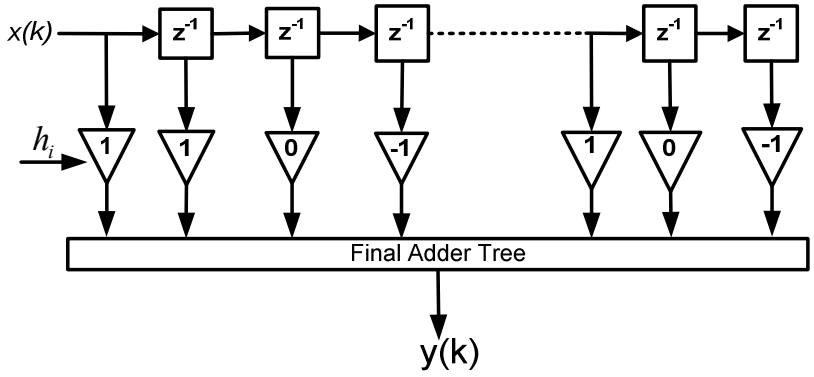


Figure 4.1 Block diagram of Ternary FIR filter (adapted from [10]).

4.3 Noise Shaping in Sigma-Delta Modulators

Quantization error is evident in any kind of ADC circuit and is produced by nonlinear behaviour of the quantization stage. This is also evident in sigma delta modulators. The quantization error can be treated as white noise so is also called quantization noise if it is uniformly distributed in the region $[+\Delta/2, -\Delta/2]$, where Δ is the quantization step size, and the input full-scale range is $[-X_{FS}/2, X_{FS}/2]$ [95]. Thus quantization error can be modelled as a white noise source, *e*, as shown in Figure 4.2. Considering an N bit ΣΔM with the number of quantization levels Q= 2^N and $\Delta = X_{FS}/Q - 1 = 2V/Q - 1$, the quantization noise can be defined as [86]:

$$\sigma^2(e) = \frac{\Delta^2}{12} = \left(\frac{2V}{2^N - 1}\right)^2 /12 \cong \left(\frac{2V}{2^N}\right)^2 /12 \qquad (4.2)$$

As quantization noise power, $\sigma^2(e)$ is uniformly distributed across the frequency range then its power spectral density (PSD) can be defined as [95]:

$$Q_E = \frac{\sigma^2(e)}{f_s} = \frac{1}{f_s}\left[\frac{1}{\Delta}\int_{-\Delta/2}^{\Delta/2} e^2 de\right] = \frac{\Delta^2}{12 f_s} \tag{4.3}$$

and the *in-band noise power*, calculated for low pass sigma-delta signals is given below:

$$P_E = \int_{-B_d}^{B_d} Q_E(f) df = \frac{\Delta^2}{12OSR} \tag{4.4}$$

where B_d is band of interest. It is evident from (4.4) that a higher oversampling ratio (OSR) can be chosen to overcome the quantization noise power at the cost of increased hardware [58] know as noise-shaping property of sigma-delta modulators. Whereas OSR is defined as the ratio of sampling frequency to the Nyquist frequency (i.e. $OSR = f_s / f_N$) and is generally chosen to be in the range of 8 to 256. Alternatively, a higher order ΣΔM may be used (i.e., a higher noise transfer function) at the cost of increased instability issues [96]. The in-band noise power for M^{th} ΣΔM is represented by the following relationship:

$$P_Q \equiv \int_{-B_d}^{B_d} Q_E \left|NTF(f)\right|^2 df \cong \frac{\Delta^2}{12} \frac{\pi^{2M}}{(2M+1)OSR^{2M+1}} \tag{4.5}$$

where $\left|NTF(f)\right|^2$ is the squared magnitude of the noise transfer function that is exploited to push the quantization noise outside the band of interest (i.e., B_d) and M is the order of sigma-delta modulator.

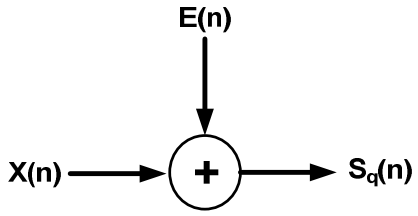


Figure 4.2 Linear Model of Quantizer

Similarly, if the signal is treated as a zero-mean random process and it is sinusoidal in nature with power σ_x^2, the Signal-to-Quantization Noise Ratio (*SQNR*) of the sigma-delta modulator can be defined as:

$$SQNR = 10\log\left(\frac{\sigma_x^2}{\sigma_q^2}\right) = 10log\left(\frac{\sigma_x^2}{V^2}\right) + 4.77 + 6.02(dB) \qquad (4.6)$$

The direct relationship between SQNR of an ADC and its resolution is clear from(4.6). Thus, each single-bit increase in the SDM quantization will add 6-dB in SQNR [86].

To be useful in generating the ternary taps, the ΣΔM must use a ternary quantizer and must possess a flat frequency response over the bandwidth of interest. In this design example we have used a second order sigma delta modulator for the generation of ternary coefficients that is presumed stable in nature than higher order sigma-delta modulators (i.e., ≥ 3) and has better SQNR than a first order modulator. A typical structure of a second order ΣΔM was used to generate the ternary coefficients as shown in Figure 3.3.

4.3.1 Ternary Filter Design in MATLAB

The design of a Ternary FIR filter in MATLAB is performed by the same way as was discussed in section 3.3.2. As a demonstration of the technique, the interpolated target impulse response of Figure 4.3 was derived for a filter with a roll-off between 800 and 1000Hz at an OSR of 32 and the ternary filter simulated at varying OSRs. It can be seen from Figure 4.4 that the corresponding impulse response closely matches the target impulse response of the ternary filter for each OSR. Also from Figure 4.4, it can be seen that each doubling of the OSR results in an increase of about 10-dB in the stop-band attenuation, starting with 20dB at 32 OSR and reaching about 50-dB at an OSR of 256. Note that the frequency axis in Figure 4.4 has been normalized to make the comparison clearer. Varying the OSR implies a different absolute sampling rate for each (e.g., $\text{Nyquist rate} \times \text{OSR}$) thus, although the relative characteristics of the filter will not change, the actual edge frequencies will be correspondingly different. A down sampling filter is used at the very end of the process to readjust the overall filter frequency responses to its original (Nyquist) rate [83].

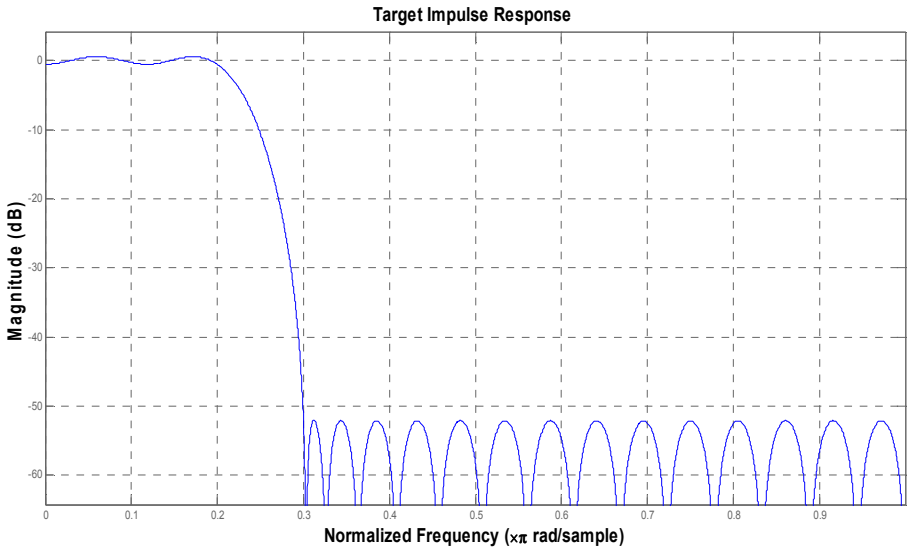


Figure 4.3 Target Impulse Response of FIR filter

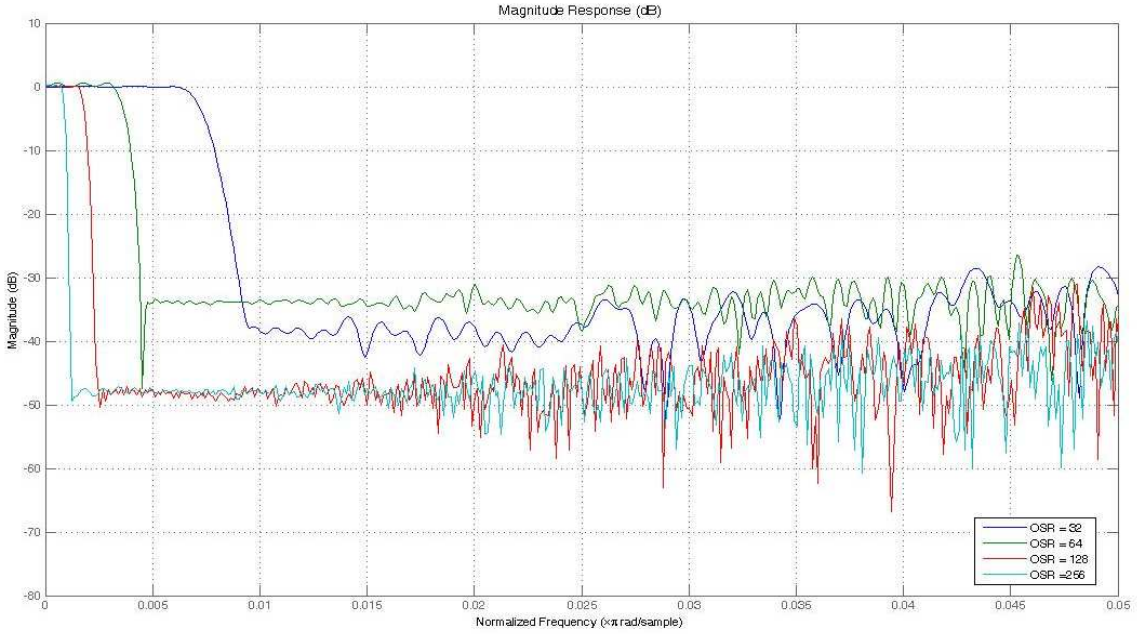


Figure 4.4 Ternary Filter Impulse Response at OSR = 32, 64,128 and 256

It can be seen that the resulting frequency response (Figure 4.5) closely matches the form of the target impulse response in Figure 4.3. This ternary filter frequency response was obtained by generating White Gaussian Noise as input to the ternary filter and computed the average 8192-points FFT over 1000 realizations with a Hanning window.

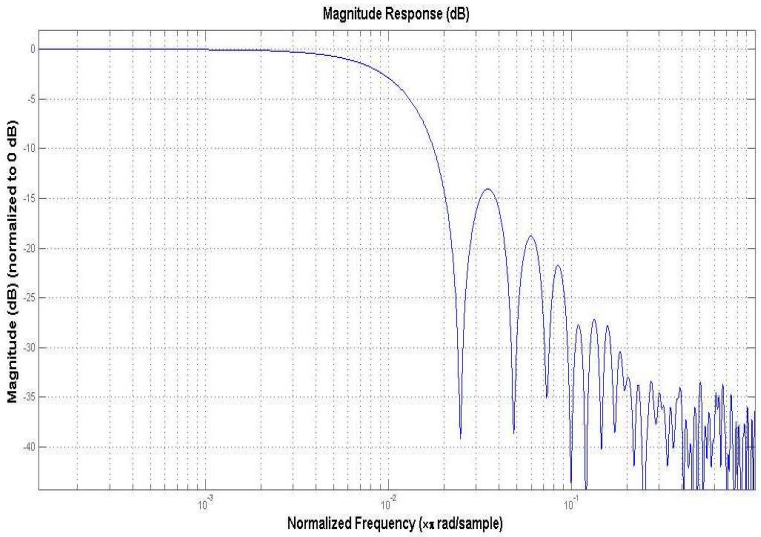


Figure 4.5 Frequency Response of a Ternary FIR Filter

4.4 Ternary FIR Filter Design in FPGA

As discussed above, the structure of a ternary FIR Filter (see Figure 4.1) comprises two main stages: the multiplication of input data by the coefficients followed by the addition of the partial products. In the speech filter of [10], the number of taps (M) could be large: 1024 for an over-sampling rate (OSR) of 32, or 2048 for an OSR of 64. Thus, in the following simulations we have chosen OSR in the range of 8 – 64 as the

ternary taps can be accommodated in the smallest available FPGA devices, for example – the Altera Cyclone-II. Hence, the number of ternary coefficients will be given by $OSR \times NR$. As the order of the Nyquist rate (NR) coefficients here is 32, the *OSR* range defined above may lead to a coefficient order in the range $2^8 \rightarrow 2^{11}$. The implementation explored in this work divides this up into *N* coefficient multiply blocks (i.e., by ±1, 0) followed by an adder tree with $log_2 N$ levels to perform the summation. In a binary implementation therefore, the addition would require 10 or 11 stages (e.g., *N*=1024). As the performance of the adder tree is critical to the filter, we have experimented with three alternative implementations—in 2's complement binary, redundant binary sign-digit representation (RBSD) of the type described in [97] and canonical signed digit (CSD) [98]. These will be described in the next section.

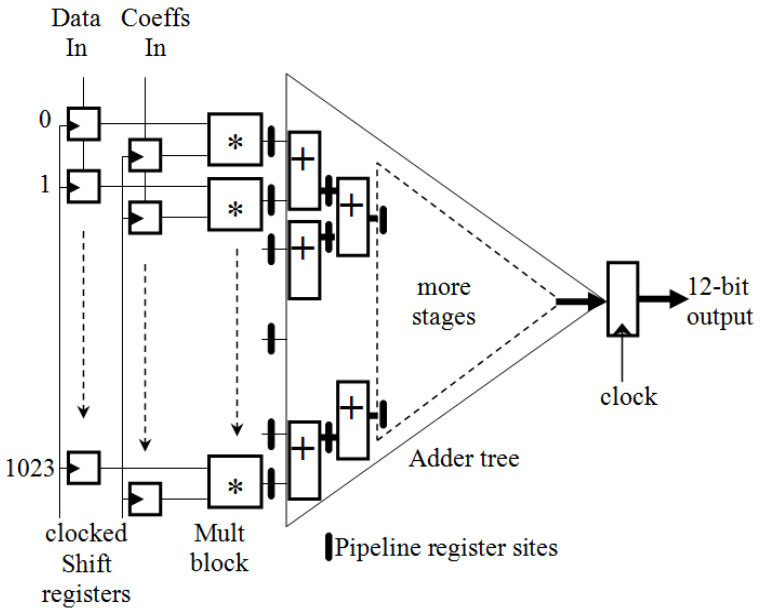


Figure 4.6 TFF hardware architecture

The general block diagram of the adder is shown in Figure 4.6. The adder tree as shown comprises ten levels (i.e., ternary taps ± 1024). At each stage of the tree, the number of addition blocks halves while their length increases by one bit, culminating in the final 12-bit output. As a typical FPGA LUT structure takes a small number of inputs (in the range of 6 to 8), the first two adder levels will be mapped to individual LUT blocks in the FPGA architecture that will operate in parallel. The remainder will comprise small ripple-carry blocks up to twelve bits long. Note that it would be equally possible to use optimised IP blocks created specifically for this purpose. Throughout this thesis, a more general approach has been taken, so that the results can be considered to be worse-case.

As a result, the overall performance of the filter will be determined primarily by the adder blocks. The following sections examine the effect of three alternative implementation strategies on the area and performance of the adder stages: 2's complement, redundant binary signed digit (RBSD) and canonical signed digit (CSD) encoding techniques. The area and performance of an example filter has been simulated in pipelined and non-pipelined modes for all three techniques.

4.4.1 Two's-complement

Ternary data and coefficients may be represented most simply as two-bit, two's-complement numbers. This has the advantage of simplifying the arithmetic as addition by either +1 or -1 becomes the same operation. Summing over N=1024 implies ten levels and a final multi-bit result of ±N. However, because the two's-complement

representation is offset around zero, 12 bits are required to completely express the full output range of ±1024. As mentioned above, a key advantage of SWL systems is that the multiplication process becomes trivial—it can be performed using a small AND-OR logic equation instead of the complex multiplication of standard multi-bit organizations. In the FPGA context, the multiplication of two 2-bit numbers will map to a small LUT, thereby saving significant area and power. A traditional Wallace adder tree structure has been used in [99], for the implementation of efficient adder and the VHDL code is generated by MATLAB. This thesis focuses on techniques that map efficiently onto FPGA structures.

4.4.2 Redundant Binary Signed Digit (RBSD) Representation

Various forms of redundant binary representations have been used for many years to achieve restricted carry or carry-free operation in adder circuits [97, 100]. Here, the Redundant Binary Signed Digit (RBSD) method of [97] has been adapted as an alternative to the 2's complement binary method described in the previous section. These methods typically exchange a fixed critical path delay for control over the propagation of the carry (the same objective as carry-lookahead, for example). The intention has been to examine how this technique might be used to balance the pipeline stages in an FPGA implementation. From even a cursory examination of the overall filter structure, it is likely that the earlier stages, with small bit lengths (e.g., in the range 3—5 bits), will exhibit longer critical path delays than their corresponding

binary circuits while the latter stages might be shorter. More importantly, carry-free operation implies that each stage should exhibit a similar propagation delay, rebalancing the pipeline and resulting in improved overall performance. However, this has proved to be true only for the smallest device examined in this study as will be identified in the following section.

X	Y	W0	T0	W1	T1	S
-1	-1	0	-1	0	-1	-1
-1	0	1	-1	-1	0	-1
-1	1	0	0	0	0	0
0	-1	1	-1	-1	0	-1
0	0	0	0	0	0	0
0	1	-1	1	1	0	1
1	-1	0	0	0	0	0
1	0	-1	1	1	0	1
1	1	0	1	0	1	1

(a) Addition logic table

0	0	X2	X1	X_0
0	0	Y2	Y1	Y_0
0	0	$W0_2$	$W0_1$	$W0_0$
0	$T0_2$	$T0_1$	$T0_0$	0
	$W1_3$	$W1_2$	$W1_1$	$W1_0$
	$T1_2$	$T1_1$	$T1_0$	0
	S_3	S_2	S_1	S_0

(b) Addition Mechanism

Figure 4.7 RBSD addition

From Figure 4.7 it can be seen that the input operands decode to two intermediate pairs of terms (*W0 and T0*) that then translate to *W1* and *T1* such that these two cannot simultaneously be equal (i.e., both -1 or +1). The carry-free addition is completed by

simply adding *W1* and *T1* at each bit position. Addition therefore requires three stages: two intermediate translations and a final add logic stage (i.e., XOR). As a result, there is a performance tradeoff between the delays incurred by carry propagation between adjacent adder blocks and the longer critical paths required by the carry-free RBSD logic. It appears likely that the availability of optimized carry chain logic within current FPGAs will tend to favour the former organization over the latter.

4.4.3 Canonical Signed Digit (CSD) Representation

Canonical Signed Digit (CSD) is an approach that has been widely adopted due to its reduced complexity and better area-performance tradeoffs [101-102]. The technique was first reported in 1960 by Reitwiesner [98] and further improved by many researchers (i.e., for example see [103]) . Using the CSD encoding technique, nonzero digits can be reduced such that the number of partial products is minimized, which can lead to efficient hardware implementation [101]. The two important properties of CSD are that no two adjacent digits are nonzero and each representation of a number is unique [98]. This implies that there are atmost *N/2* non-zero digits for an *N*-bit number, in contrast to the equivalent 2's complement number which can have up to *N* non-zero digits. A given 2's complement number, *a*, can be represented in CSD encoding by the following relationship:

$$a = -a^n 2^n - 1 + \sum_{i=0}^{n-2} a_i 2^i = \sum_{i=0}^{n-1} c_i 2^i \qquad (4.7)$$

where $a_i \in (0,1)$ and $c_i \in (-1,0,+1)$. The ternary nature of the canonical system therefore results in a number with fewer non-zero digits than 2's complement. Additionally, in contrast to the 50% probability that a digit is zero in 2s complement, a n-bit CSD number exhibits a probability of 2/3 that an individual bit will be zero, which tends to 1/3 as n becomes larger as given by the relationship [94]:

$$P(|c_i| = 1) = 1/3 + (1/9n)\left[1 - (-1/2)^n\right] \tag{4.8}$$

These characteristics make CSD highly useful for general purpose DSP applications, especially filters. For example the four bit two's complement representation of 15 is (1111), which exhibits four non-zero digits. The CSD encoding of the same number is $\bar{1}0001$ with only two non-zero digits. Similarly, the CSD encoding of $-25 = 00\bar{1}0100\bar{1}$ contrasts with the 2's complement representation of $-25 = 11100111$. The smaller number of non-zero digits achieved using the CSD encoding simplifies the multiplication process such that that it can be realized with shifters, adders and subtracters.

The basic principle of CSD multiplication is quite straightforward. The multiplicand is either added to or subtracted from the accumulator depending on the least significant bit (LSB) of the multiplier e.g. (+1, -1). No operation is required in the case of a zero bit. The accumulator is then right shifted by one-bit. This is performed from the LSB of the multiplier until the most significant bit (MSB). The right shift

operation of the accumulator has to perform with sign-extension from its MSB. Finally, the accumulator must be cleared before a new multiplication is started.

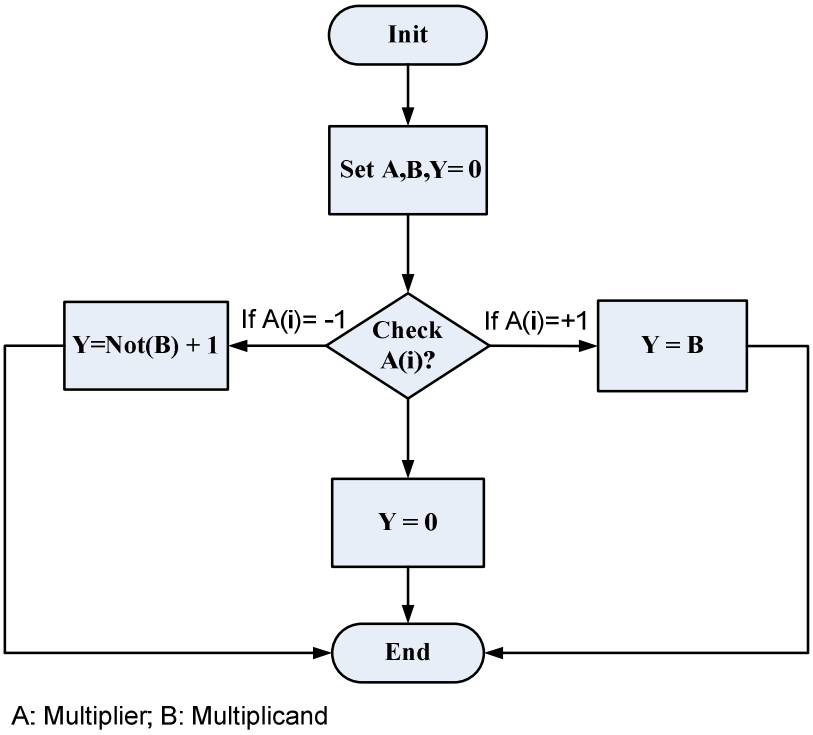


Figure 4.8 Flow chart of the single-bit ternary CSD Multiplier

In the case of a single-bit ternary encoding, CSD multiplication becomes trivial. The CSD multiplier shown in Figure 4.8 requires only add or subtract operation. There is no need for a subsequent shift operation because the operands are single-bit. Moreover, as the multiplicand and the multiplier are in a single-bit format, the product is a single-bit as well. This can be mapped efficiently to a small LUT in the FPGA. This hardware simplicity offers a distinct advantage to single-bit CSD FIR filter implementations.

4.5 Simulation Results and Discussion

The three filter design alternatives were coded in VHDL and compiled, simulated and synthesized using Quartus-II 9.1 and Modelsim 5.8 targeting a small number of Altera™ Cyclone and Stratix FPGAs, chosen as representative of a range of FPGA devices from low-cost to through to high performance. The adder tree was synthesized in both pipelined and non-pipelined configurations. In the pipelined mode, registers are used between each adder stage as illustrated in general terms in Figure 4.6. As in all pipelined systems, the tradeoffs involve a small increase in hardware area and greater latency in exchange for increased throughput. It can be noted that the latency of the adder stage (log_2N+1) is small compared with the overall data latency across the filter shift registers, but its impact will be entirely application dependent.

Similar to the chapter 3, in these entire tests the target operating frequency (F_{MAX}) was set at a value higher than was known to be achievable for the given technology, to force the optimization tools to generate a near best-case P&R that is more comparable across the range of devices. The approximate area values listed are those reported by the flow summary.

Table 4.1 reports the non-pipelined mode area-performance results obtained for all three filter designs. The RBSD scheme operates on 2-bit digits so that the bit width increases by two at each successive level in the adder tree compared with one extra bit per stage in its 2's complement counterpart. This imposes a significant area overhead

on the filter, almost doubling the LUT usage in most cases. Clearly, a combination of the additional complexity of the RBSD system and the resulting increase in routing delay (due to a greater number of register and LUTs blocks) overwhelms any advantage to be gained from the lack of carry propagation. Typically, the two's complement approach results in an average performance between 35% and 39% better than with RBSD encoding, as shown in Table 4.1. In contrast, a CSD representation that offers significant area savings and performance enhancement within a multi-bit context has negligible benefit in the ternary domain.

Table 4.1. Area-Performance Results in Non-pipelined mode Two's complement, RBSD, and CSD coefficients results

Non-Pipelined Mode Results with three representing techniques							
Device	Ternary Taps	Two's complement		RBSD		CSD coefficients	
		LUTs	F_{MAX}	LUTs	F_{MAX}	LUTs	F_{MAX}
Cyclone – II EP2C70F896C6	256	2412 (4%)	78.28	4604 (7%)	51	2433 (4%)	81.4
	512	4875 (7%)	66.27	9203 13%)	45	4845 (7%)	68.6
	1024	10,108 (15%)	61.5	18498 (27%)	38.6	9977 (15%)	61.89
	2048	20188 (30%)	52.55	37298 (55%)	33.66	20032 (29%)	51.78
Stratix – II EP2S180F1508 C3	256	1882 (1%)	101.42	3660 (3%)	64.5	1738 (1%)	101.93
	512	3800 (3%)	91.83	7475 (5%)	56	3659 (3%)	91.22
	1024	7591 (5%)	78.65	14862 (10%)	49.2	7322 (5%)	80.79
	2048	14999 (10%)	69.68	29631 (21%)	43	14385 (10%)	68.93

Almost identical area-performance results were recorded compared to the simple 2's complement approach in both non-pipelined and pipelined modes (Table 4.1 and

Table 4.2). It is likely that the small area-performance differences obtained are the result of differences in the fitment or place and route processes, the assigned frequency constraints and/ or the particular routing paths selected by the synthesis tools for a that synthesis run.

Interestingly, RBSD outperforms 2's complement in pipelined mode with Cyclone-II device but exhibits comparatively poor performance using the Stratix-II. The very large change seen in the RBSD area-performance in pipelined mode is most likely to be due to the different architectures of the FPGA devices. The primary difference between these two devices is their basic configurable logic unit. The Cyclone series is made up of 4-input LUT units while Stratix series devices include a more flexible technique known as the Adaptive Logic Module (ALM) in which the basic LUT can be flexibly partitioned into sub-blocks with different numbers of inputs. Typically, adder tree based designs of this type are able take advantage of the additional partitioning opportunities offered by this ALM organization.

In our case, RBSD performs well with Cyclone-II compared to Stratix-II in pipelined mode because the basic 4-input LUT of the former is fully utilized by the design, resulting in shorter paths that reduce the overall delay between stages (Table 4.3). In this table, the input bit column describes the bit widths of successive adder stage and the number of LUTs represents the minimum number required to accommodate that bit width. It can be seen that RBSD occupies a full LUT at each stage compared to 2's complement with the Cyclone-II device.

Table 4.2. Area-Performance tradeoffs in pipelined mode

Pipelined Mode Simulation Results							
Device	Ternary Taps	Two's Complement		RBSD		CSD coefficients	
		LUTs	FMAX	LUTs	FMAX	LUTs	FMAX
Cyclone – II	256	2884 (4%)	223	5057 (7%)	304	2888 (4%)	220.5
	512	5806 (8%)	210.5	10235 (15%)	281	5794 (8%)	208
	1024	11628 (17%)	187	20558 (30%)	274	11620 (17%)	186.22
	2048	23400 (34%)	169	41255 (60%)	253	23389 (34%)	172.5
Stratix – II	256	1753 (1%)	423	3525 (2%)	356	1755 (1%)	415
	512	3561 (2%)	415.15	7118 (5%)	346.4	3551 (2%)	413.1
	1024	7119 (5%)	381.97	14269 (10%)	331.9	7119 (5%)	381.9
	2048	14280 (10%)	350	28587 (20%)	297.5	14280 (10%)	350

In contrast to the case with multi-bit systems, the CSD technique offers no significant advantage over 2's complement in pipelined mode. In a similar way to non-pipelined mode, again 2's complement has an average 16% better performance than RBSD technique with Stratix-II device. However, an average 30% better performance is achieved by RBSD with Cyclone-II device as compared to 2's complement approach.

4.6 Summary

In this chapter three alternative encoding techniques were investigated in FPGA namely: 2's complement, RBSD, and CSD. All three techniques were synthesized and

simulated using small commercial FPGA devices in pipelined and non-pipelined modes. It was found that in the small adder blocks that form this filter (<12 bits) there is little, if any, advantage in using a non-binary representation or canonical signed digit. Languages such as VHDL do not automatically access optimal resources such as fast carry hardware so that these performance results can be considered to be worse-case. Thus, we can see that ΣΔ modulation-based FIR filter can achieve high performance while requiring neither special attention to carry propagation nor the use of built-in DSP components such as fast parallel multipliers making the technique well suited to ASIC implementation.

Table 4.3. Two devices LUTs requirement in 2's complement and RBSD approach

Stage #	2's complement			RBSD		
	Input bits	Number of LUTs		Input bits	Number of LUTs	
		Cyclone-II	Stratix-II		Cyclone-II	Stratix-II
1	2	1	1	2	1	1
2	3	2	1	4	2	2
3	4	2	1	6	3	2
4	5	3	2	8	4	2
5	6	3	2	10	5	3
6	7	4	2	12	6	3
7	8	4	2	14	7	4
8	9	5	3	16	8	4
9	10	5	3	18	9	5
10	11	6	3	20	10	6

In next chapter mathematical model of novel narrowband single-bit ternary adaptive channel equalization model is proposed. This model has been simulated in

MATLAB and verified according to Symbol error rate and minimum mean squared error (MMSE).

Chapter – 5

Single-bit Ternary Adaptive Channel Equalization for Narrowband Signals

5.1 Introduction

The sole purpose of an adequate adaptive equalizer is to combat Inter-symbol-interface (ISI) in a dispersive channel [104]. Alternatively it is described as the channel effects that appear in the received signal due to the low pass nature of the channel. Various works are reported to develop hardware efficient, accurate and faster adaptive channel equalization algorithms [105]. It is no surprise that channel equalizers are the backbone of all means of communication especially wireless communication. A high data rate with multiple channel transmission is a core demand of wireless communication that can be achieved with an appropriate equalization algorithm depending upon the circumstances and environmental conditions.

Equalization techniques for band-limited timer dispersive channel may be subdivided into two types – linear equalization and non-linear equalization [106-107]. With both type of equalizers an adaptive algorithms are associated with typical structural styles. Figure 5.1 shows an overview of equalizers types, structures and algorithms. Typically, transversal and lattice are two basic structures to implement any equalization technique. Transversal structure is $(N-1)$ tapped delay line multiplication with coefficients format that is easy to implement in hardware. However, lattice structure comprises of feed-forward and feedback filters instead of only feed-forward. It has a complex structure but gives better performance compared to the transversal structure.

A linear equalizer may be implemented as a FIR filter with adjustable coefficients. Usually these coefficients are adjusted according to information provided and error computation at the end of the detector as show in Figure 5.2. Linear equalizers are easy to implement and find use in applications where the channel distortion is not too severe [105]. Varieties of adaptive algorithms are available to develop a linear equalizer like: LMS (whole family), RLS, Fast RLS, Square-Root RLS and Gradient RLS. All these algorithms can be implemented with transversal structure except gradient RLS. LMS family algorithm can be categorized as LMS, block LMS (BLMS), sign LMS (SLMS), normalized LMS (NLMS), and their further derivatives. Zero forced equalization (ZFE) is also a linear approach to remove all the ISI from the received signal without taking care of the SNR i.e., noise enhancement problem.

Minimum Mean Square Error (MMSE) is another approach to optimize the equalizer filter by reducing the affects of ISI and taking care of SNR as well [108]. The later approach is widely accepted and used in communication due to its robustness and immunity to noise enhancement. Equalization can be symbol-to-symbol or sequence based.

A major problem with linear equalizers is the presence of nulls in the channel impulse response. Due to these nulls at some point the bandwidth of the input signal $x(t)$ will cause noise power $n(t)$ become infinite. This problem greatly affects the signal to noise ratio (SNR) and ultimately prevents the equalizer from converging, eliminating the equalization characteristics. In this scenario, non-linear equalizers are required that are less prone to the effect of deterioration of the SNR when spectrum nulls are found in channel impulse response but at the same time are not too complex and non-linear.

Decision Feedback Equalizer (DFE), Maximum a *posteriori* probability (MAP) and Maximum Likelihood Sequence Estimation (MLSE) are three major non-linear equalization methods[109]. There are tradeoffs among all three techniques in the sense of complexity and performance. MLSE is the optimum equalization technique that minimizes the probability of a sequence error that can be implemented by Viterbi algorithm [109]. However, the computational complexity of MLSE increases exponentially with number of symbols affected by ISI [105]. In terms of complexity, the decision feedback equalizer is far better than MLSE [110] and same is true for

MLSE on the other parameter. Unlike the linear equalizer (where error is computed by the difference of expected output and observed output (i.e., $e = d - \hat{d}$)), in this case a decision device is used for the computation of an error. Inclusions of decision device and feedback filters are major reasons of complexity and instability in non-linear filters.

Training and Blind modes are two approaches to mitigate the ISI affects. In the training mode the overhead of training pulses are send to receiver well before the actual transmission that is loss of bandwidth. Blind equalization doesn't need such transmission but bit complex than former technique [111].

In all these algorithms, and techniques channel information is very much important to equalize the channel impulse response with tap-weights (i.e., equalization filter). Despite of all this, all these algorithms are developed using multi-bit domain except sign algorithm that adopts input, or coefficient, or error as sign term instead of multi-bit domain[112] to resolve the hardware complexity. However, in some way it has multi-bit domain aspects.

Recently, a novel single-bit adaptive algorithm has been proposed to suppress noise in narrowband signals using sigma delta modulated filters [5, 14]. In this filter, the primary inputs, internal signals, adaptive filter coefficients, error term and final output all are in a single-bit format. In [82], it was shown that ternary FIR filters implemented in FPGAs using pipelined and non-pipelined organizations could exhibit superior

performance compared to their multi-bit counterparts. Thus, single-bit DSP systems have the tendency to reduce the multiplier complexity that gives better performance with comparable chip area.

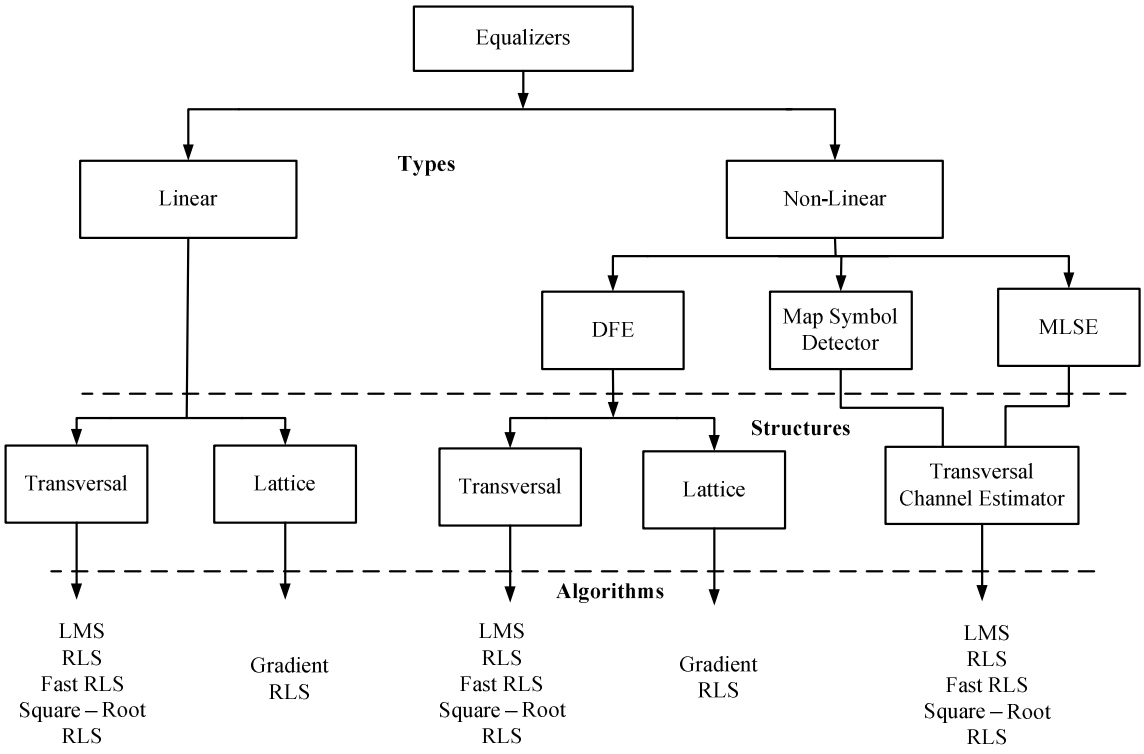


Figure 5.1. Equalizers types, structures, and algorithms [106]

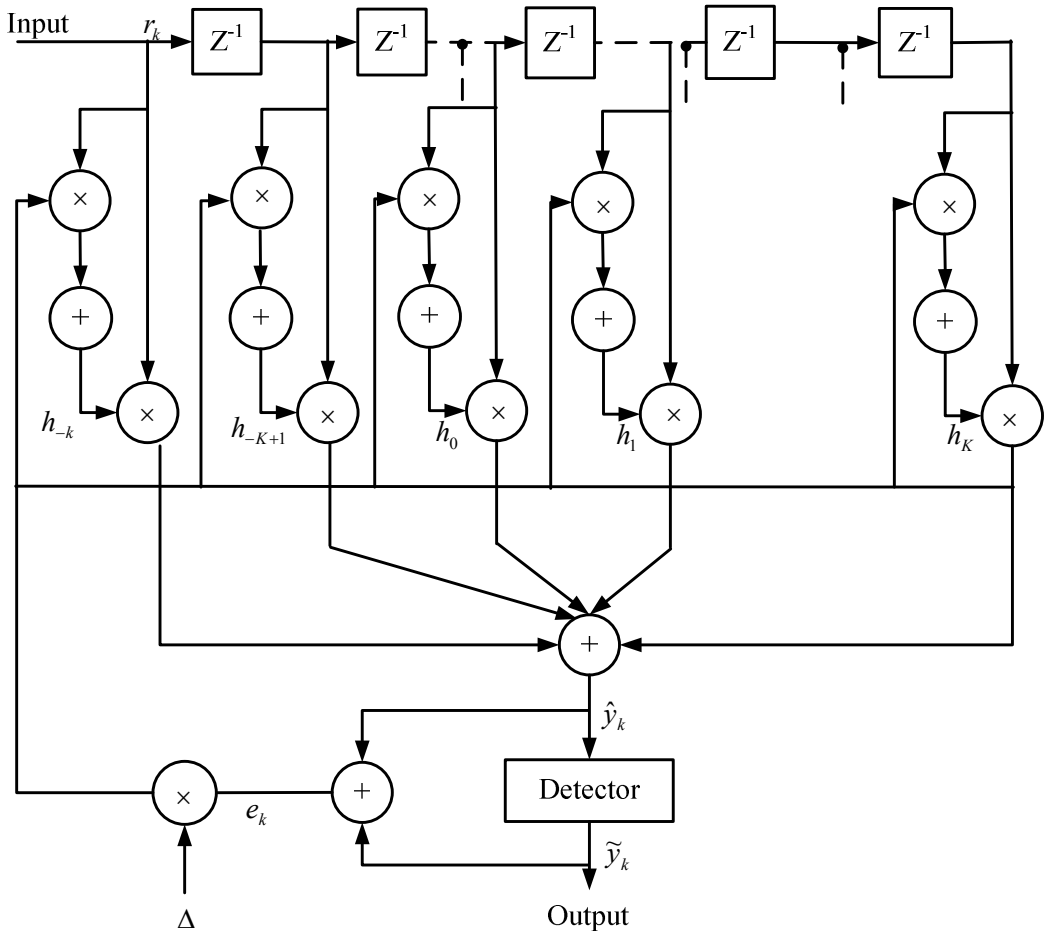


Figure 5.2. Adaptive linear FIR equalizer with LMS algorithms [106]

In this chapter a novel design of single-bit adaptive channel equalization is proposed based on my publication [113]. In this design, the multi-bit channel equalization model used in contemporary communication systems (see Figure 5.3) is modified with a novel single-bit model (see Figure 5.4) [114]. In this model, the received signal, equalization coefficients, and final output all are in single-bit domain. A delayed version of the input is used as the desired input signal. In this work, the

input narrowband signal is oversampled and converted into single-bit format by passing it through a sigma-delta modulator then finally modulated for transmission. The sign function was used to extract the samples {+1, -1} in MATLAB. This oversampled signal is shaped by the channel impairments (i.e., ISI effects) that change the signal format from single-bit into the multi-bit domain. So that channel equalization may be performed in single-bit domain, the received signal is passed through a ΣΔM block. In this way the overall system remains in single-bit domain and there is no need to convert between the multi-bit and single-bit formats at the transmitter and receiver stages.

In this design, a block LMS (BLMS) approach has been adopted to derive a mathematical model of single-bit adaptive algorithm. Moreover, a *single-bit* term has been used while the format of coefficients for adaptive algorithm were used in ternary (i.e., +1, 0, -1) to suppress the inter symbol affects (ISI) and noise disturbances at receiver end (see in section 5.3).

5.2 System Design

The transmitted narrow band input signal in channel equalization can be described as [111]:

$$x(n) = \sum_{p=0}^{\infty} d_p g_T(n - pT) \tag{5.1}$$

where $g_T(n)$ is the basic pulse shape that is selected to control the spectral characteristics of the transmitted signal, d_p is the sequence of the transmitted information symbols from a signal constellation consisting of *M* points, and *T* the signal interval: (*1/T*) is symbol rate[111]. In this case, pulse shaping filter is not taken into account for the sake of simplicity so the transmitted input sequence is:

$$x(n) = d_n \tag{5.2}$$

The transmitted signal is shaped by the inter-symbol interference (ISI) due to channel impairments and additive noise that can be mathematically represented as:

$$r(n) = \alpha.\left(\sum_{i=0}^{N-1} h_{ci}x(n-i) + v(n)\right) \tag{5.3}$$

where h_{ci} represents the channel impulse response and $v(n)$ is additive noise, considered to be white Gaussian noise with zero mean and variance δ^2 This received signal is passed through a second order sigma delta modulator to convert the input into a binary $\{+1,-1\}$ as shown in Figure 5.4. However, to maintain the dynamic range (DR) of the second order sigma delta modulator i.e., $\{+1,-1\}$, a gain parameter of α has been introduced to ensure that the convolution sum stays within the prescribed dynamic range.

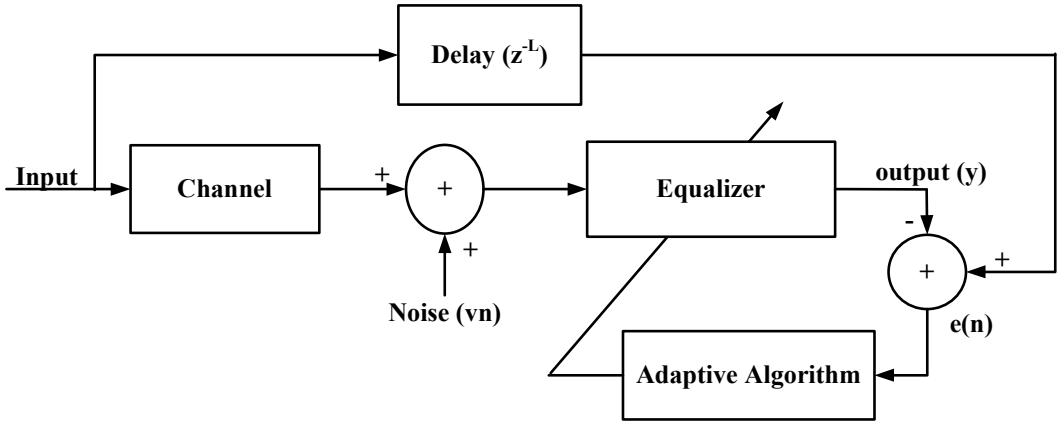


Figure 5.3. General block diagram of an adaptive equalizer

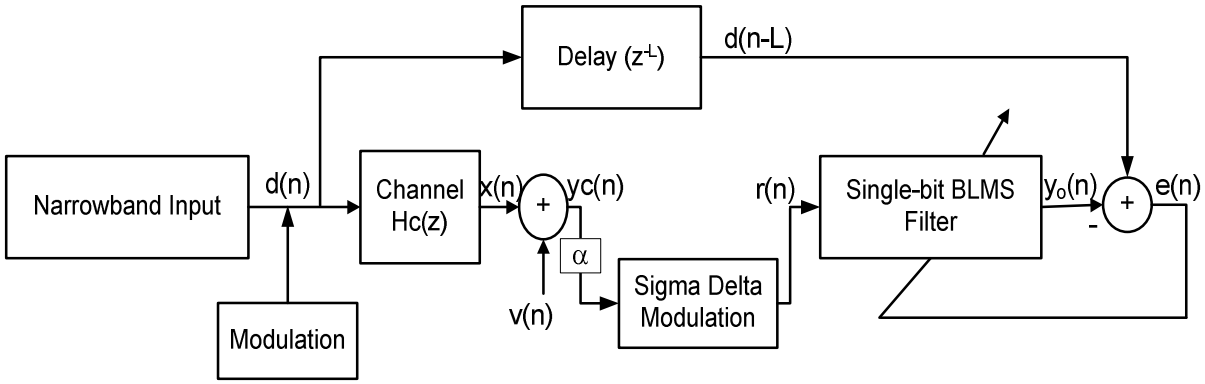


Figure 5.4. Block Diagram of Single-bit ternary Adaptive Channel Equalization

Similarly, the estimated output of the input and the equalizer coefficients is in multi-bit format (see Figure 5.6). To transform this output into the single-bit domain, a second order sigma delta modulator is introduced (Figure 5.5). However, the dynamic range of the second order sigma delta modulator that results in the best SNR, and assures the overall stability of the system is $\{+1,-1\}$. To ensure stability, a gain

parameter β is introduced as shown in Figure 5.6. Adjustment of this factor is not trivial. It varies the noise performance by varying the input level (given in section 5.3). The precise value of β can be achieved by using any adaptive SDM, such as that reported in [115].

5.3 Single-bit Ternary LMS-like Adaptive Channel Equalization Algorithm

It is already mentioned that the primary objective of the channel equalization filter is to mitigate inter symbol interference (ISI) effects in the dispersive channels and increase the output SNR. In this work, a Block LMS (BLMS) approach has been adopted to design a new single-bit ternary BLMS algorithm (SBTLMS) due to its ability to adjust more samples compared to the LMS algorithm while maintaining equivalent performance with small computational complexity [116]. Moreover, Minimum Mean Square Error (MMSE) method is used to adapt the ternary coefficients that are widely used for linear algorithms except ZFE.

The proposed SBTLMS can be derived using a standard Block LMS (BLMS) algorithm. The general block diagram of the SBTLMS is shown in Figure 5.6. The gradient approximation will be presented before moving to the derivation of SBTLMS algorithm that is based upon a similar approach.

5.3.1 Wiener Theory and LMS algorithm

Wiener proposed an adaptive filter theory that adjusts the weight(s) to produce filter output $y(n)$, which when subtracted from the corrupted signal $d(n)$ would result the output $e(n)$ a clean signal. In a standard adaptive algorithm, for a single tap an FIR filter output $y(n)$ and error signal $e(n)$ may be described as[19-20, 117]:

$$y(n) = wx(n) \tag{5.4}$$

The error term is then defined as:

$$e(n) = d(n) - wx(n) \tag{5.5}$$

where $d(n)$ is the desired response. To solve for the best weight approximation i.e., w^* starts with the taking square root of the output error while considering that desired response and observed signal (or approximation) is, in a broad sense, a stationary (WSS) process. This leads to:

$$e^2(n) = (d(n) - wx(n))^2 = d^2(n) - 2d(n)wx(n) + w^2x^2(n) \tag{5.6}$$

Taking the statistical expectation of this result, then we have:

$$E(e^2(n)) = E(d^2(n)) - 2wE(d(n)x(n)) + w^2E(x^2(n)) \tag{5.7}$$

From this equation, the statistical terms can be defined as:

$$
\begin{aligned}
&J = E(e2(n)) = \text{MSE (mean squared error)} \\
&\sigma^2 = E(d^2(n)) = \text{Power of corrupted signal} \\
&P = E(d(n)x(n)) = \text{Cross-correlation between } d(n) \text{ and } x(n) \\
&R = E(x^2(n)) = \text{Autocorrelation}
\end{aligned}
\tag{5.8}
$$

Thus the mean squared error can be described as:

$$J = \sigma^2 - 2wP + w^2R \tag{5.9}$$

Since σ^2, P, and R are constants, J is a quadratic function of w. It can be seen that J is a hyperparaboloidal surface which never goes negative. The best weight (optimal) w^* is at the location where the minimum MSE J_{min} is achieved. Taking a derivative of the cost function and setting it to zero should lead towards the optimum tap value(s):

$$\frac{dJ}{dw} = -2P + 2wR = 0 \tag{5.10}$$

Solving (5.10), we get the optimum weight solution:

$$w^* = R^{-1}P \tag{5.11}$$

Practically, prior knowledge of autocorrelation (P) and cross correlation functions (R) are not available. To overcome this issue, a LMS algorithm was described by Widrow that uses a steepest descent algorithm to minimize the MSE sample by sample and locate the filter coefficient(s). The algorithm can be described as:

$$w_{n+1} = w_n - \mu \frac{dJ}{dw} \tag{5.12}$$

where μ is the step-size that controls the rate of convergence and dJ/dw is a gradient vector at time index *n*. This relationship(5.12), shows that the weight vector is proportional to the negative gradient. The sample based processing of LMS algorithms needs instantaneous gradient descent estimation of the weight vector by taking statistical expectation out of *J* and then computing the derivative to obtain an approximate of dJ/dw:

$$\begin{aligned} J &= e^2(n) = (d(n) - wx(n))^2 \\ \frac{dJ}{dw} &= 2(d(n) - wx(n)) \frac{d(d(n) - wx(n))}{dw} = -2e(n)x(n) \end{aligned} \tag{5.13}$$

Substituting dJ/dw in the steepest descent algorithm of (5.12), the well known Widrow-Hoff LMS algorithm can be obtained:

$$w_{n+1} = w_n + \mu e(n)x(n) \tag{5.14}$$

The same approach has been taken into account to reach a single-bit adaptive solution by considering that $e(n) \in \{+1, 0, -1\}$, and $w_n \{+1, 0, -1\}$ that leads to the gradient function:

$$\frac{dJ}{dw} = -2e(n) \tag{5.15}$$

which can be then used to reach an approximation to the optimal Wiener solution in single-bit domain adaptive filter.

5.3.2 SBTLMS algorithm derivation

The input to the SBLMS is received signal that is channel impaired and Gaussian noise corrupted signal, *r(n)*. This signal is further sigma-delta modulated to covert it into single-bit format. The $N\times1$ single-bit input signal vector at the time index n can be expressed as:

$$r(n) = [r(n), r(n-1), ..., r(n-N+1)]^T \tag{5.16}$$

where $[.]^T$ indicates transposition, N represents the interpolated single-bit filter order, n is the multi-bit channel filter order. The $N\times1$ single-bit coefficient vector of the equalization filter h at time index n can be denoted by:

$$h(n) = [h_o(n), h_1(n), ..., h_{N-1}(n)]^T \tag{5.17}$$

and the single-bit estimation output vector at time n as:

$$y_o(n) = [y_o(n), y_o(n-1), ..., y_o(n-N+1)]^T \tag{5.18}$$

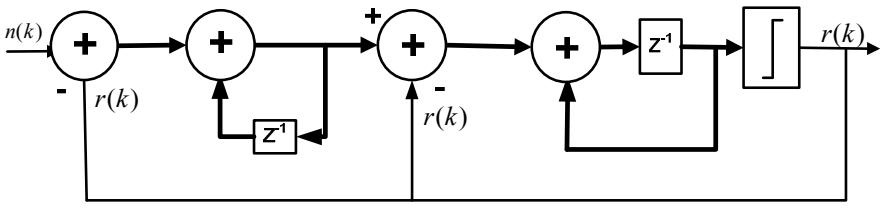


Figure 5.5. Second order sigma delta modulator

The index j refers to the block index of the block LMS algorithm which is related to the original sampling index n as follows:

$$n = j\Delta + i,\ \text{j} = 1,2,3,4,5,\ .....;\ \text{i} = 0,1,2,.....,\Delta\text{-}1 \qquad (5.19)$$

where Δ denotes the block length and i is the block index, j the number of blocks index so that $j = n/\Delta$. The LMS algorithm is a special case of the BLMS where the block length is 1. Generally, block length is considered with reference to the order of the filter i.e., Δ >N, Δ <N, or Δ =N. In general, the second and third cases are preferred to the first. In this work, second case i.e., Δ <N has been used. Additionally, due to the higher order of the single-bit equalization filter it is convenient to consider Δ and the filter order (N) in the power-of 2.

The $N \times \Delta$ single-bit input data for block j is therefore defined by the set $[r(j\Delta + i)]_{i=0}^{\Delta-1}$, which can be expressed in matrix form as:

$$U(j) = [r(j\Delta), r(j\Delta + 1), ..., r(j\Delta + \Delta - 1)] \qquad (5.20)$$

The tap weight vector *h(j)* remains constant over this block of input data. The estimated output of this filter, $\{(\hat{r}(j\Delta + 1)\}$ produced by the equalization filter in response to the input signal vector $r(j\Delta + i)$ is given by:

$$\hat{r}(j\Delta + 1) = h^T(j)\ r(j\Delta + i) \qquad (5.21)$$

However, this expected output is the result of a convolution operation between single-bit input samples and single-bit coefficients so the output generated at this stage

should be in multi-bit format. To keep the entire system within the single-bit domain this output is passed through the second order sigma delta modulation (Figure 5.5).

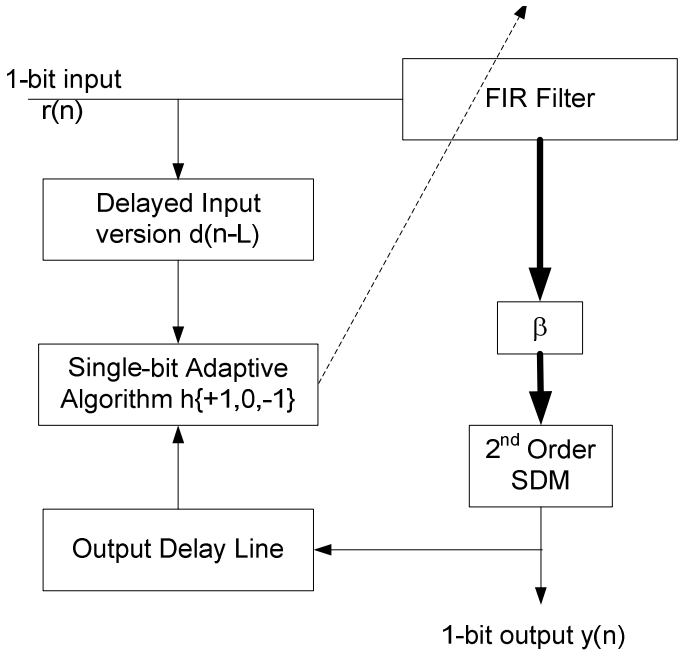


Figure 5.6. General block diagram of single-bit block LMS-like filter [5]

As the dynamic range of the second order sigma delta modulator should be in the range of $\{+1,-1\}$ to achieve the best SNR, a scale factor is used to maintain this range. An important measure of the SDM is to keep flat signal frequency response over the desired band of the frequency.

Thus, the SDM of the expected signal should not modify the specifications of the estimated output. The single-bit version of the expected output can be described as:

$$y_o(j\Delta + i) = \text{sgn}[(\beta \hat{r}(j\Delta + i)] \tag{5.22}$$

where β is a scaling factor and the sgn can be described as below:

$$\mathrm{sgn}(\delta) = \begin{cases} +1 & \delta > 0 \\ 0 & \delta = 0 \\ -1 & \delta < 0 \end{cases}$$

The second order sigma delta modulator used here has the following transfer function:

$$H(z) = S(z)z^{-1} + E(z)(1 - 2z^{-1} + z^{-2}) \tag{5.23}$$

where $S(z)$ represents the signal transfer function and $E(z)$ the quantization noise transfer functions. The noise shaping effect of the ΣΔM is evident from the presence of the filtering term, $(1 - 2z^{-1} + z^{-2})$ acting on the noise term, $E(z)$. This quantization effect of the sigma delta modulator can easily be approximated by using a linear approximation [25]. Therefore the expected output with quantization noise shaping can be expressed as:

$$y_o(j\Delta + i) = \beta \hat{r}(j\Delta + i) + q_{yo}(j\Delta + i) \tag{5.24}$$

where q_{yo} represents the shaped quantization noise due to the modulation affect that is generated in the response to the convolution between noise impulse response coefficients and block of the quantization noise. Hence $\Delta \times 1$ quantization noise vector can be defined as:

$$q_{yo}(j) = [q_{yo}(j\Delta), q_{yo}(j\Delta + 1), ..., q_{yo}(j\Delta + \Delta - 1)]^T \tag{5.25}$$

Thus, the single-bit output can be expressed as:

$$y_o(j\Delta + i) = \beta h^T(j) r(j\Delta + i) + q_{yo}(j\Delta + i) \tag{5.26}$$

or in matrix form as:

$$y_o(k) = \beta\ U^T(j) h(j) + q_{yo}(j) \tag{5.27}$$

where $U(j)$ is $N \times \Delta$ size matrix that can be generated using the Toeplitz built-in function in MATLAB or by exploiting the matrix format.

In single-bit domain error term is accounted into coefficient update formula that is similar to the multi-bit block LMS (BLMS) algorithm. The error is simply considered to be the desired signal (i.e., d_{n-L}) subtracted from the expected output (y_o), defined in block terms as:

$$e(j\Delta + i) = d_{n-L}(j\Delta + i) - y_o(j\Delta + i) \tag{5.28}$$

In simple form the error is:

$$e(j) = d_{n-L}(j) - y_o(j) \tag{5.29}$$

Now the weights update formula for the single-bit domain can be described as:

$$h(j+1) = \text{sgn}[h(j) + mu \times e(j)] \tag{5.30}$$

where mu is the controlling factor and $h(j)$ are the coefficients in the range of $\{+1,\ 0, -1\}$, that is the function of the ternary quantizer. In this work, we have used (5.30) relationship for the tap weight vector update.

As shown in section 5.1, here, we have drawn the transversal structure of SBTLMS filter using the relationship (5.30) as shown in Figure 5.7. In this structure, error is computed as difference between delayed version of transmitted signal and expected output (shown in Figure 5.4) but shown notionally with same input just to avoid another line.

The ternary format of the coefficients results a harsh quantization affect (i.e., it introduces quantization noise) that can be expressed by using linear approximation as shown previously. Unlike multi-bit BLMS, here, averaging terms of input and error [116] are not considered because single-bit nature of the system will not add any further improvement by including these terms. Therefore, the updating function can be approximated as:

$$h(j+1) = h(j) + mu \times [r(j) - y_o(j)] + q_w(j) \tag{5.31}$$

where the quantization noise $q_w(j)$ is a $N \times 1$ vector. Thus updating function becomes:

$$\begin{aligned} h(j+1) = h(j) + mu \times [r(j) - \\ \beta U^T(j) h(j) + q_{yo}(j)] + q_w(j) \end{aligned} \tag{5.32}$$

In these equations, all the single-bit adaptive process parameters and quantization error components are given.

5.4 Simulation and discussion

In this work, the single-bit ternary block LMS-like (SBTLMS) algorithm has been simulated using narrowband input signal in MATLAB. For this simulation, a narrowband 11-tap low pass filter channel model (Hc) was selected to create an equivalent channel equalizer filter (He) in single-bit domain.

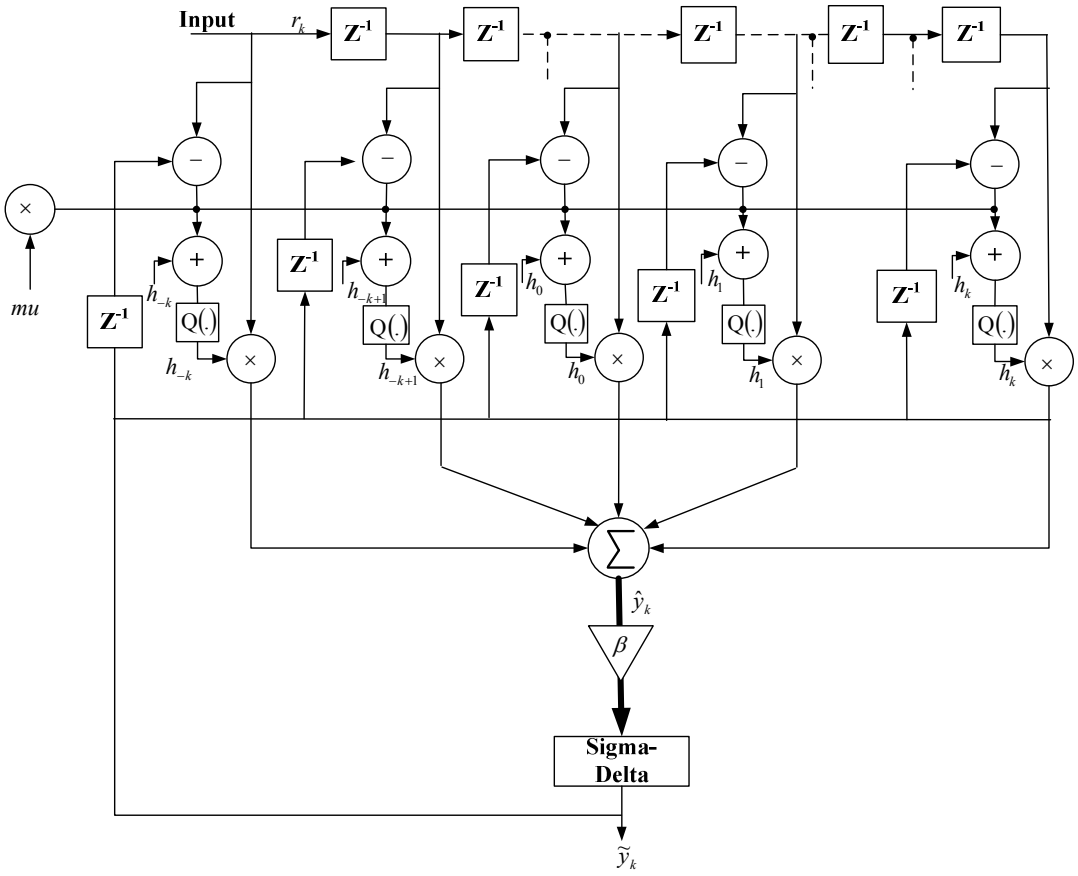


Figure 5.7. The proposed SBLMS adaptive algorithm structure

In these tests, the narrowband input signal $d(n)$ was chosen half of the maximum voice frequency (i.e., $f_o = 2000Hz$) with an amplitude of $A = 0.5$. It was assumed that input signal $d(n)$ is an oversampled single-bit signal throughout the simulation. The Nyquist rate of the channel filter order was selected as $N=11$, , and the oversampling ratio was chosen as OSR=128 and the equivalent filter order was defined by using the relationship $(OSR \times N)$.

5.4.1 Symbol Error Rate (SER) at Varying input Training Samples

Initially, SER was calculated using varying input training samples that were recorded in decision directed mode. Hence, SDM oversampled input was filtered through the oversampled equalized channel filter model (He). However, in the single-bit domain it is not trivial to find the starting point due to delays introduced by the channel impairments. Thus, initial 100 samples were discarded to reach a starting point that has a small SER value. The SER is shown at various input training samples in Figure 5.8. In a subsequent stage, the SER was recorded at varying SNR as shown in Figure 5.9.

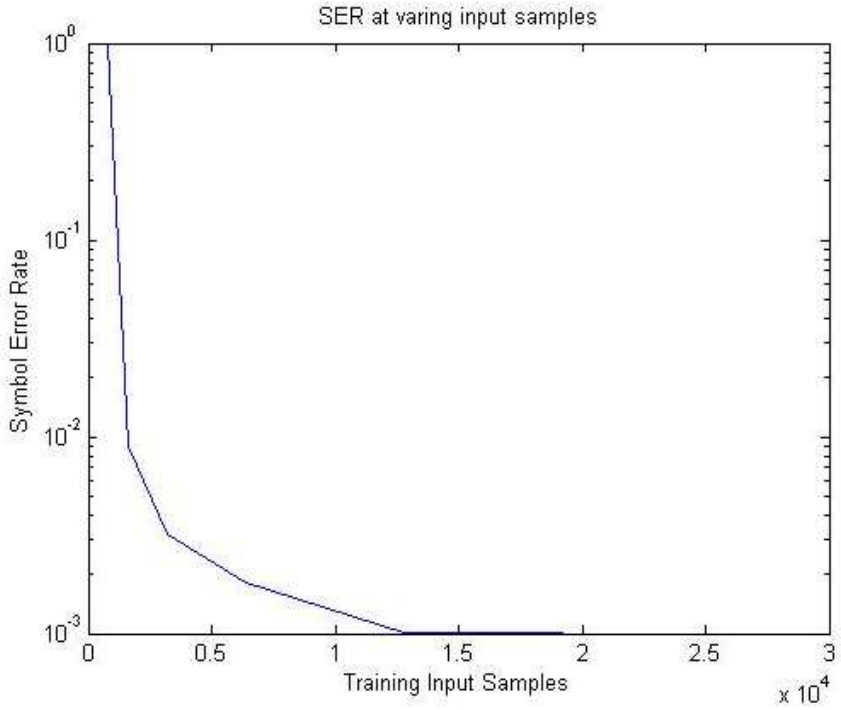


Figure 5.8. SER at varying input training samples

5.4.2 Signal-to-Noise Ratio (SNR)

The improvement in the SNR was treated as a measure of the performance and was defined as the output divided by the input (in-band) SNR. An improvement in the output SNR was recorded at varying input SNR. However, the dynamic range of the second order sigma delta modulator that results in the best SNR, and assures the overall stability of the system is $\{+1,-1\}$. To ensure stability, a gain parameter β is introduced as shown in Figure 5.5. Considering the non- negative values of the expected output in the range $(1, N)$ then this factor may be defined as:

$$\frac{2}{N} < \beta < 2 \qquad (33)$$

The precise value of β can be achieved by using any adaptive SDM, such as are reported in [115]. Thus an appropriate β factor was set to achieve the best SNR while keeping the in-band frequency same. Extensive simulations indicate an optimum around *OSR*6.7*. Simulations repeated under the same conditions sometimes show different performance due to the noise and ISI that continuously change the gain factor and therefore the dynamic range.

In these simulation results, the best performance achieved has been considered at OSR=128 and input sinusoid at $fo = 2000Hz$. It is evident that the best performance is achievable at the full dynamic range of the expected output ($\hat{x}$) that is ± 1. These results are given in Table 5.1.

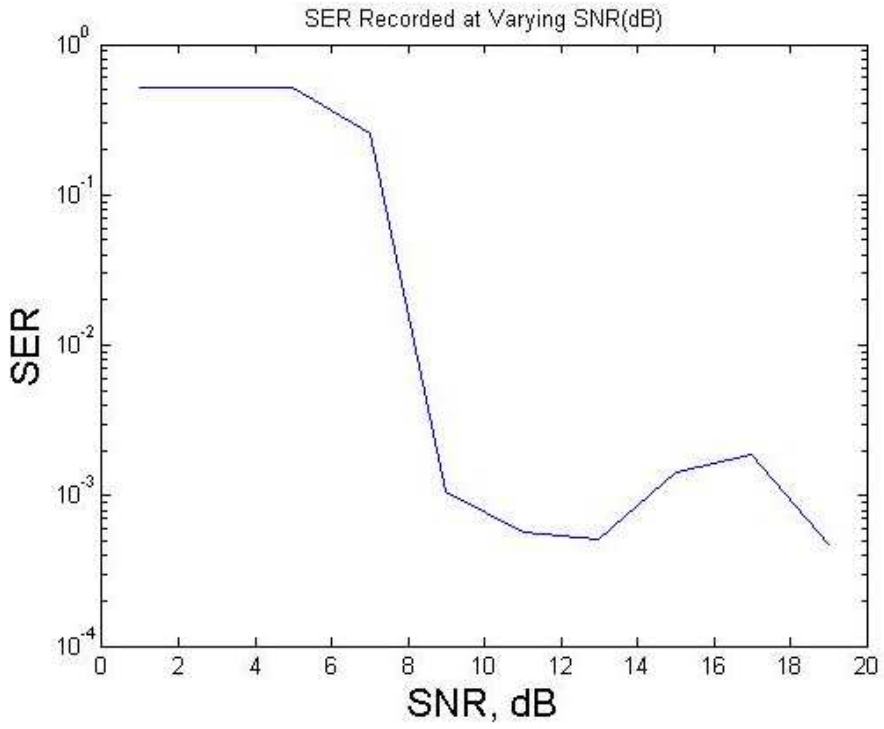


Figure 5.9. SER recorded at varying input SNR(dB)

5.4.3 Minimum Mean Squared Error (MMSE)

The performance in the LMS or BLMS filters is measured in terms of its Minimum Mean Squared Error (MMSE). In single-bit systems, this error is not a continuous function but is bounded within the range $\{+2,0,-2\}$, which makes it harder to determine the gradient analytically. However, the mean squared error may be determined in the same way as current adaptive algorithms. In this way, an ensemble average learning curve of the sample can be defined as:

$$P(j\Delta + i) = E\left|d(j\Delta + i) - yo(j\Delta + i)\right|^2 \tag{5.34}$$

where E denotes the expectation operator. The ensemble-average learning curve over the interval of $0 \le j \le N$ is defined as the average over the O trials as:

$$\hat{P}(j\Delta + i) = \frac{1}{O}\sum_{l=1}^{O}\left|e_l(j\Delta + i)\right|^2 \tag{5.35}$$

where $\hat{P}(j\Delta + i)$ is the sample-average approximation of the actual learning curve. The desired response here is the delayed version of the input signal *d*. In this work, $\hat{P}$ has been derived using (23) averaged over a number of trials from 1 to 30 on an input signal with additive Gaussian noise. It is evident from Figure 5.10 that the MMSE is trending towards a small final value around zero.

5.5 Summary

In this chapter, a novel single-bit adaptive channel equalization model for narrowband input signals has been proposed. The overall system is kept within the single-bit domain including input signal, filter taps (ternary format), final output and error terms. A narrow band low pass filter channel model was selected to demonstrate the proposed model simulation results in MATLAB. The model exhibits significant results in the sense of SER at varying training input pulses and at varying SNR. The MMSE was shown to trend towards zero. Improvement in the SNR was recorded at varying in-band input SNR. The model results in low hardware complexity, especially in FPGA devices.

Table 5.1 Improvement in the SNR_o recorded with varying input SNR_i

No.	β	SNR (dB)	Pndb (dB)	SNRo (dB)
1	OSR*6.7	2	-11.03	17.810
2	OSR*6.7	3	-12.03	17.44
3	OSR*6.7	4	13.03	17.35
4	OSR*6.7	5	-14.03	17.315
5	OSR*6.7	6	-15.03	17.99
6	OSR*6.7	7	-16.03	18.21
7	OSR*6.7	8	-17.03	17.637
8	OSR*6.7	9	-18.03	19.43
9	OSR*6.7	10	-19.03	19.48
10	OSR*6.7	15	-24.03	16.3

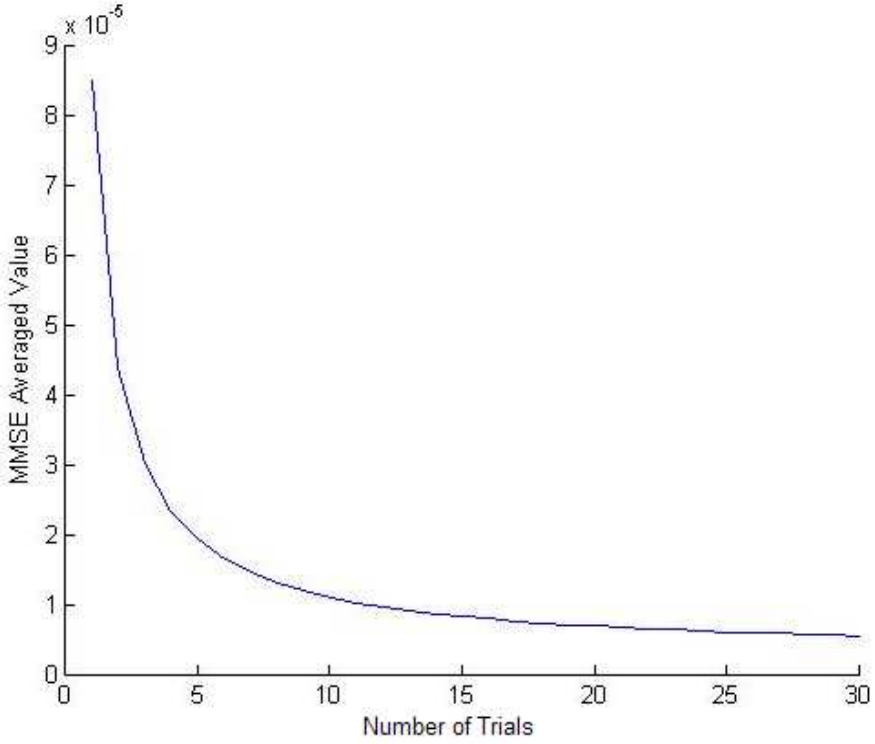


Figure 5.10. MMSE averaged over 1 to 30 trials

Chapter – 6

Conclusion and Future Directions

6.1 Introduction

This thesis can be divided into two sections. In first section we have presented the analysis and synthesis of single-bit ternary DSP algorithms on small commercial FPGA devices. These experimental results are compared to the approximately equivalent multi-bit system. It is shown that single-bit techniques typically achieve better area-performance tradeoffs compared to its corresponding part multi-bit system. In a second stage, we have studied the stability of single-bit ternary FIR filter and proposed a new design of single-bit ternary adaptive channel equalization for narrowband signals.

In chapter 3, an FIR-like $\Sigma\Delta$ modulator filter was synthesized on small commercial FPGA devices. Both the input and output of this filter are in short word length format (i.e., single-bit, ternary). The coefficients of this filter (the ternary taps) were generated in MATLAB by selecting a target impulse response using the Remez exchange algorithm. This filter comprises of two sections: the ternary filter and an IIR re-modulator filter. The latter section is used to remove the quantization noise and bring back the ternary filter multi-bit output in single-bit format so overall system remains in single-bit domain.

To analyse the area-performance-power tradeoffs results between contemporary (i.e., multi-bit) and single-bit techniques both filters were synthesized and simulated in pipelined and non-pipelined modes at roughly equivalent levels of spectral performance using 2's complement encoding. This equivalent spectral performance level was determined at a theoretical as well a practical level.

The dynamic power simulations were conducted in two stages. Firstly, both the filters were simulated at their maximum clock frequency determined by the worse-case F_{MAX} for either the single-bit or the multi-bit filter, related via the performance of the filter. In a second step, the two filter types were set up to achieve the specifications outlined for specific application at Nyquist rate. Similarly, a single-bit filter was set to achieve an equivalent spectral performance to the multi-bit case by making corresponding OSR, so in this case the single-bit filter clock was obtained by

multiplying the OSR to the Nyquist frequency (i.e., F_S*OSR). The multi-bit filter clock was kept at its Nyquist rate (8000Hz) throughout.

FPGA based investigations have shown that a single-bit FIR-like filter using ternary coefficients will routinely dissipate less power compared to the conventional approach despite their need to operate at much higher clock rates. They also exhibit up to 40% higher performance and offer useful area savings at lower filter orders. At higher orders, the ΣΔ approach retains its power and performance advantages but exhibits slightly higher chip area. The simplicity and low power of the ΣΔ approach makes it applicable to mobile communication processing using low cost FPGA or ASIC technology.

In this investigation two important factors were found; firstly, the IIR filter is a limiting factor for single-bit ternary filter performance. A stand-alone ternary filter can achieve clock frequency higher than 400 MHz but it was restricted to 250 MHz with re-modulator IIR filter section. In pipelined mode, due to complex multiplication the multi-bit filter performance degrades about 30% whereas single-bit filter performance is almost un-changed.

The stability of sigma-delta modulation based systems is of great concern due to non-linear behaviour of the circuit. Although Sigma-Delta modulators have been applied as re-modulator blocks in single-bit filter system, their inherently non-linear behaviour leads to stability concerns. The primary sources of this nonlinearity are the

1-bit quantizer, op-amp slew rate, op-amp nonlinear DC gain, and nonlinear switch response. The most important of these is the non-linear behaviour of the quantizer that may very easily lead to the instability of the sigma delta modulator. When applied in simple systems such as a first order IIR filter loop, positive feedback within the loop causes the system to become unstable after just a few input samples.

In continuation of SBTFF comparison to its counterpart multi-bit system, here, we have attempted to analyse the stability of single-bit ternary FIR filter and proposed a new model that takes into account widely accepted input and quantizer gain limits of the second order sigma delta modulator inside the IIR filter. A linear analysis of the limits to the quantizer gain and feedback parameter is presented.

Investigation shows that proposed method gives sufficient control over the input to the sigma delta modulator and the quantizer gain while improving the acceptable limits of IIR filter feedback parameter. With this new design, the IIR loop feedback parameter has been relaxed and can increase to 1.5 without compromising the stability of the system. The gain function adds little to the complexity of the filter, ensuring it can be efficiently mapped to hardware with fewer data bits, saving chip area and improving the system performance.

The investigation in Chapter 3 was performed using 2's complement approach. However, in multi-bit systems canonical signed digit (CSD) encoding is often considered better than 2's complement due to its lower multiplication complexity that

can achieve high throughput. Hence, in Chapter 4, the ternary FIR filter was further investigated to find the area-performance tradeoff at three different classical encoding techniques called 2's complement, canonical signed digit (CSD), and redundant binary signed digit (RBSD). Ternary filter was synthesized and simulated at various order of the filter and OSR on small commercial FPGA devices in Quartus-II in non-pipelined and pipelined modes.

An investigation into the three encoding techniques found that, unlike in the equivalent multi-bit filters, CSD offers no advantages to single-bit sigma-delta modulated systems. Similarly, RBSD occupies twice the area and exhibits much poorer performance compared to a conventional 2's complement representation due to the small symbol size in single-bit systems. These results demonstrate that simple, short word-length ΣΔM filters will be useful in greatly reducing the number of general-purpose digital multipliers in general purpose DSP applications using Field Programmable Gate Arrays (FPGA) and especially ASIC.

In chapter 5, a new design of single-bit adaptive channel equalization is proposed using sigma delta modulation and a single-bit ternary block Least Mean Square (SBTLMS) algorithm. The overall system is kept within the single-bit domain including input, filter taps, final output and error terms. A narrow band low pass filter channel model was selected to demonstrate the model.

With correlated narrowband input signals, this model was investigated and found that it is able to converge and to form an equalization filter with good SNR and very low Symbol Error Rate (SER). As the input, filter coefficients and output values are all in single-bit and/or ternary format its overall hardware complexity will be be low compared to traditional multi-bit channel equalization. Additionally, the technique avoids the need for successive conversion from multi-bit to single bit and back at the receiver and transmitter stages. Moreover, it opens a door to potential new research in the area of adaptive filters that may lead towards less complex and highly efficient DSP systems offering high throughput for important applications such as adaptive channel equalization.

6.2 Future Directions

While every effort has been made in this thesis to cover the relevant topic as thoroughly as possible, inevitable time constraints have prevented potentially interesting investigations into different various optimization techniques, structures and improved designs. In this section we present a brief discussion on some topics found in this thesis that would prove useful to be investigated further.

1. Design and synthesis of single-bit ternary FIR-like filter with direct form structure achieved better area-performance-power characteristics than its counterpart multi-bit is one way of implementation in FPGA. There are many optimized structures reported in literature to design FIR filter [102]. Further

optimization of the single-bit ternary FIR-like filter in FPGAs can improve the area-performance-power tradeoffs compared to multi-bit system. In particular, the efficient implementation of the IIR re-modulator stage would have an immediate impact upon the performance of the overall system. There are many ways to potentially achieve this goal, which would be a large investigation [29].

2. Further investigation is required to analyse and understand the stability limitations of short word length DSP algorithms. Many recent publications have proposed non-linear stability analysis of sigma-delta modulation [35, 92] that may further be taken into account for the analysis of single-bit systems. This investigation may lead us to the commercial product that could change the contemporary systems with simple and more effective single-bit designs.

3. Higher order sigma-delta modulation organizations should be investigated for area-performance-power characteristics of single-bit ternary FIR-like filter in FPGA that may lead towards lower chip area and higher performance at the cost of bit higher probability of instability.

4. Three encoding techniques (i.e., CSD, RBSD, 2's complement) needs more customized investigation that could explore that how CSD ternary encoding affects SBTFF, and what would be an impact of 4-input adder with small bit-width as compared to the traditional 2-input adder?

5. Single-bit DSP algorithms design should be synthesized and investigated in Application Specific Integrated Circuit (especially using commercial EDA tools such as Cadence) to better determine the area, performance and power tradeoffs in that domain. It is already clear that ASIC designs could utilise the SWL approach proposed in this theses. It would be useful to explore whether the tradeoffs that exist in the ASIC designs are similar to those observed in the FPGA environment. This would also lead the SWL technology closer to commercial production.

6. Single-bit adaptive LMS-like algorithm derived for channel equalization can further be investigated using more complex channel characteristics and random nature of the input signal. This investigation may be useful to understand the behaviour of single-bit algorithms in different environments. Other important factor is its convergence rate that is controlled by a factor called '*mu*'. In single-bit format it is limited to few values (i.e.,>05, <0.5, or = 0.5). Further investigation is required to understand the '*mu*' factor that impacts upon the convergence rate and ultimately Minimum Mean Square Error (MMSE).

7. Single-bit adaptive algorithm can further be extended towards its FPGA design and analysis using binary or ternary format of coefficients and compare it with LMS (or block LMS) algorithms with area-performance characteristics and its real-time operation. This would be a large investigation to understand how SWL designs can be accommodated in current mobile communication.

REFERENCES

[1] C. Cheng and K. K. Parhi, "Low-Cost Parallel FIR Filter Structure with 2-Stage Parallelism," *IEEE Transaction on Circuits and Systems,* vol. 54, pp. 280-290, 2007.

[2] P. K. Meher, S. Chandrasekaran, and A. Amira, "FPGA Realization of FIR Filter by Efficient and Flexible Systolization Using Distributed Arithmetic," *IEEE Transaction on Signal Processing* vol. 56 pp. 3009-3017, Jul. 2008.

[3] R. A. Hawley, B. C. Wong, T.-j. Lin, J. Laskowski, and H. Samueli, "Design Techniques for Silicon Compiler Implementations of High-Speed FIR Digital Filters," *IEEE Journal of Solid-State Cicuits,* vol. 31, pp. 656-667, May 1996.

[4] S. Shanthala and S. Y. Kulkarni, "High Speed and Low Power FPGA Implementation of FIR Filter for DSP Applications," *European Journal of Scientific Research,* vol. 31, No.1, pp. 19-28, 2009 2009.

[5] Amin Z. Sadik and Z. M. Hussain, "Short Word-Length LMS Filtering " in *ISSPA 2007*, Sharjah, UAE, 2007, pp. 1-4.

[6] D. Johns and D. Lewis, "IIR filtering on sigma-delta modulated signals," *Electronics Letters,* vol. 27, pp. 307-308, 1991.

[7] C. W. Ng, N. Wong, and T. S. Ng, "Bit-stream adders and multipliers for tri-level sigma–delta modulators," *IEEE Transactions on Circuits and Systems II: Express Briefs,* vol. 54, pp. 1082-1086, 2007.

[8] A. C. Thompson and Z. M. Hussain, "All Digital Adaptive Sigma-Delta Modulator," in *Proc. WITSP'04*, 2004.

[9] A. C. Thompson, Z. M. Hussain, and P. O'Shea, "Efficient Digital Single-Bit Resonator," *IEEE Electronic Letters* vol. 40, pp. 1396-1397, 2004.

[10] A. C. Thompson, P. O'Shea, Z. M. Hussain, and B. R. Steele, "Efficient Single-Bit Ternary Digital Filtering Using Sigma-Delta Modulator," *IEEE Letters on Signal Processing,* vol. 11, pp. 164-166, February 2004.

[11] A. Z. Sadik, Z. M. Hussain, and P. O'Shea, "A single-Bit Digital DC-Blocker Using Ternary Filtering," in *Proc. Tencon'05*, 2005, pp. pp:1-6.

[12] A. Z. Sadik, Z. M. Hussain, and P. O'Shea, "Structure for single-bit digital comb filtering," in *Proc. APCC'05*, 2005, pp. 545 - 548.

[13] A. C. Thompson, Z. M. Hussain, and P. O'Shea, "A Single-Bit Narrow-band Bandpass Digital Filter," *Institute of Engineers Australia (IEAUST) Electronic Journal* pp. 31-40, 2005.

[14] A. Z. Sadik, Z. M. Hussain, and P. O'Shea, "An adaptive algorithm for ternary filtering," *IEE Electronics Letters,* vol. 42, pp. 420 - 420, March 2006.

[15] P. Beckett and T. D. Memon, "Reconfigurable Blocks Based on Balanced Ternary " *Journal of Signal Processing Systems,* 2010.

[16] N. Benvenuto, L. E. Franks, and F. S. H. Jr, "Realization of finite impulse response filters using coefficients +1, 0, and -1," *IEEE Transactions on Communications,* vol. COMM-33, pp. pp: 1117- 1125 October 1985.

[17] S. S. Abeysekera and K. Padhi, "Design of multiplier free FIR filters using a LADF Sigma-Delta modulator," in *Proceedsing of IEEE International Symposium on Circuits and Systems, 2000* Geneva, 2000, pp. 65-68 vol. 2.

[18] N. Benvenuto, L. Franks, and F. Hill Jr, "Dynamic programming methods for designing FIR filters using coefficients-1, 0 and+ 1," *Acoustics, Speech and Signal Processing, IEEE Transactions on,* vol. 34, pp. 785-792, 1986.

[19] B. Widrow and M. Kamenetsky, "On the statistical efficiency of the LMS family of adaptive algorithms," *Proceedings of the International Joint Conference on Neural Networks 2003, Vols 1-4,* pp. 2872-2880, 2003.

[20] B. Widrow, M. Lehr, F. Beaufays, E. Wan, and M. Bilello, "Adaptive Signal-Processing," *Wcnn'93 - Portland, World Congress on Neural Networks, Vol Iv,* pp. 548-558, 1993.

[21] B. Widrow, J. M. Mccool, M. G. Larimore, and C. R. Johnson, "Stationary and Nonstationary Learning Characteristics of Lms Adaptive Filter," *Proceedings of the Ieee,* vol. 64, pp. 1151-1162, 1976.

[22] C. S. Gunturk, J. C. Lagarias, and V. A. Vaishampayan, "On the robust of single-loop sigma-delta modulation," *IEEE Transactions on Information Theory,* vol. 47, pp. 1735-1744, July, 2001.

[23] I. Galton, "Delta-Sigma Data Conversion in Wireless Transceivers," *IEEE Transaction on Microwave Theory and Techniques,* vol. 50, pp. 302 - 315, Jan. 2002.

[24] S. S. Abeysekera and C. Charoensak, "Efficient Realization of Sigma-Delta (Sigma-Delta) Kalman Lowpass Filter in Hardware Using FPGA," *EURASIP journal on applied signal processing,* vol. 9, p. 52736, 2006.

[25] A. M. Pervez, H. V. Sorensen, and J. V. D. Spiegel, "An Overview of Sigma-Delta Converters," *IEEE Signal Processing Magazine,* pp. 61-84, Jan. 1996.

[26] H. Fujisaka, M. Sakamoto, and M. Morisue, "Bit-stream signal processing circuits and their application," *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences,* vol. 85, pp. 853-860, 2002.

[27] X. Wu, "One-bit processing for wireless networked real-time control," in *48 IEEE Conference on Decisio and Control and 28th Chinese Control Conference*, Shanghai, P.R. China, Dec. 2009, pp. 2023-2027.
[28] A. C. Thompson, Z. M. Hussain, and P. O'Shea, "A Single-Bit Digital Non-Coherent Baseband BFSK Demodulator," in *Proc. TENCON'04*, 2004, pp. pp:515- 518
[29] A. Sadik and P. O'Shea, "Realization of ternary sigma-delta modulated arithmetic processing modules," *EURASIP Journal on Advances in Signal Processing,* vol. 2009, pp. 1-7, 2009.
[30] Y. Murahashi, H. Hotta, S. Doki, and S. Okuma, "Pulsed Neural Networks Based on Delta-Sigma Modulation with GHA Learning Rule and Their Hardware Implementation," *The IEICE Transactions on Information and Systems,* pp. 705-715, 2004.
[31] R. Schreier and G. C. Temes, *Understanding delta-sigma data converters*: IEEE press New Jersey, 2005.
[32] A. Tabatabaei and B. A. Wooley, "A two-path bandpass sigma-delta modulator with extended noise shaping," *IEEE Journal of Solid-State Circuits,* vol. 35, pp. 1799-1809, 2000.
[33] R. W. Adams and R. Schreier, "Stability Theory for Sigma Delta Modulators," in *Delta-Sigma Data Convertes: Theory, Design, and Simulations*, ed: IEEE Press, 1997.
[34] R. T. Baird and T. S. Fiez, "Stability analysis of high-order delta-sigma modulation for ADC's," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing,* vol. 41, pp. 59-62, 1994.
[35] S. H. Ardalan and J. P.Paulos, "An Analysis of Nonlinear Behaviour in Delta-Sigma Modulators," *IEEE Transactions on Circuits and Systems,* vol. CAS-34, 1987.
[36] S. Kershaw, S. Summerfield, M. Sandler, and M. Anderson, "Realisation and implementation of a sigma-delta bitstream FIR filter," *IEE Proc.- Circuits, Devices, and Systems,* vol. 143, pp. 267-273, 1996.
[37] R. Schreier, "An Empirical-Study of High-Order Single-Bit Delta-Sigma Modulators," *Ieee Transactions on Circuits and Systems Ii-Analog and Digital Signal Processing,* vol. 40, pp. 461-466, Aug 1993.
[38] S. Ghanekar, S. Tantaratana, and L. E. Franks, "Design and architecture of multiplier-free FIR filters using periodically time-varying ternary coefficients," *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on,* vol. 40, pp. 364-370, 1993.
[39] D. A. Johns and D. M. Lewis, "Design and analysis of delta-sigma based IIR filters," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing,* vol. 40, pp. 233-240, 1993.
[40] C. Dick and F. Harris, "High-performance FPGA filters using sigma-delta modulation encoding," in *IEEE International Confernece on*

Acoustics, Speech, and Signal Processing, ICASSP, 1999, pp. 2123-2126.

[41] S. S. Abeysekera and X. Yao, "Optimum Laguerre filter design technique for sigma-delta demodulators," 2000, pp. 405-408 vol. 5.

[42] C. Dick and F. Harris, "FPGA signal processing using sigma-delta modulation," *IEEE Signal Processing Magazine,* vol. 17, pp. 20-35, Jan. 2000.

[43] P. W. Wong, "Fully sigma-delta modulation encoded FIR filters," *IEEE Transactions on Signal Processing,* vol. 40, pp. 1605-1610, June 1992.

[44] B.Steele and P.O'Shea, "Design of ternary digital filters," in *Procedings of the third International Conference on Information, Communication and Signal Processing (ICICS)*, Singapore, Oct. 2001.

[45] P. W. Wong and R. M. Gray, "FIR filters with sigma-delta modulation encoding," *IEEE Transaction on Acoustics, Speech, and Signal Processing,* vol. 38, pp. 979-990, 1990.

[46] C. Dick and F. Harris, "Narrow-band FIR filtering with FPGAs using sigma-delta modulation encoding," *The Journal of VLSI Signal Processing,* vol. 14, pp. 265-282, 1996.

[47] C. L. Chen and A. Willson Jr, "Higher order sigma-delta modulation encoding for design of multiplierless FIR filters," *IET Electronics Letters,* vol. 34, pp. 2298-2300, 1998.

[48] C.-W. Ng, N. Wong, and T.-S. Ng, "Bit-Stream Adder and Multiplier for Tri-Level Sigma-Delta Modulators," *IEEE Transaction on Circuits and Systems-II: Express Breifs,* vol. 54, pp. 1082-1086, Dec., 2007 2007.

[49] C. W. Ng, N. Wong, H. K. H. So, and T. S. Ng, "On IIR based bit stream multipliers," *International Journal of Circuit Theory and Applications,* vol. 39, pp. 149-158, 2011.

[50] J. Candy, "Decimation for sigma delta modulation," *Communications, IEEE Transactions on,* vol. 34, pp. 72-76, 1986.

[51] S. R. Powell and P. M. Chau, "Efficient narrowband FIR and IFIR filters based on powers-of-two sigma-delta coefficient truncation," *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on,* vol. 41, pp. 497-505, 1994.

[52] T. D. Memon, P. Beckett, and Z. M. Hussain, "Design and Implementation of Ternary FIR filter using Sigma Delta Modulation," in *Proc. ISCCC'09*, 2009, pp. 169-173.

[53] P. W. Wong, "Fully sigma-delta modulation encoded FIR filters," *IEEE Transactions on Signal Processing,* vol. 40, pp. 1605-1610, June 1992.

[54] A. C. Thompson, "Techniques in Single-Bit Digital Filtering," RMIT University, 2004.

[55] A. C. Thompson and Z. M. Hussain, "A Single-bit Resonator," in *Proc. WITSP'04*, 2004.

[56] S. M. Kershaw, S. Summerfieid, and M. B. Sandler, "On signal processing remodulator complexity," in *IEEE Internatinoal Symposium on Circuits and Systems (ISCAS)*, 1995, pp. 881-884

[57] D. Johns, W. Snelgrove, and A. Sedra, "Adaptive recursive state-space filters using a gradient-based algorithm," *IEEE Transaction on Circuits and Systems,* vol. 37, pp. 673-684, 1990.

[58] A. C. Thompson, Z. M. Hussain, and P. O'Shea, "Performance of a new single-bit ternary filtering system," in *Proc. ATNAC'03*, 2003.

[59] A. Sadik, Z. Hussain, and P. O'Shea, "Adaptive algorithm for ternary filtering," *Electronics Letters,* vol. 42, pp. 420-421, 2006.

[60] H. Fujisaka, R. Kurata, M. Sakamoto, and M. Morisue, "Bit-stream signal processing and its application to communication systems," *IEE Proc.- Circuits, Devices, and Systems,* vol. 149, pp. 159-166, 2002.

[61] C. Ng, N. Wong, and T. Ng, "Efficient FPGA implementation of bit-stream multipliers," *Electronics Letters,* vol. 43, pp. 496-497, 2007.

[62] C. Ng, N. Wong, and T. Ng, "Quad-level bit-stream adders and multipliers with efficient FPGA implementation," *Electronics Letters,* vol. 44, pp. 722-724, 2008.

[63] C. H. Dick and F. J. Harris, "Direct digital synthesis: some options for FPGA implementation," in *Reconfigurable Technology: FPGAs for Computing and Applications* 1999, p. 2.

[64] T. Katao, K. Hayashi, T. Fujisaka, T. Kamio, and K. Haeiwa, "Sorter-based sigma-delta domain arithmetic circuits," in *18th European Conference on Circuit Theory and Design*, 2007, pp. 679-682.

[65] R. S. Grover, W. Shang, and Q. Li, "A faster distributed arithmetic architecture for FPGAs," in *FPGA'02*, Monterey, California, USA, 2002, pp. 31-39.

[66] H. Yoo and D. V. Anderson, "Hardware-efficient distributed arithmetic architecture for high-order digital filters " in *IEEE International Conference on Acoustics, Speech, Signal Processing (ICASSP)*, Mar 2005, pp. 125-128.

[67] Y. Jang and S. Yang, "Low-power CSD linear phase FIR filter structure using vertical common sub-expression," *Electronics Letters,* vol. 38, pp. 777-779, Jul 18 2002.

[68] A. G. Dempster and M. D. Macleod, "Use of minimum adder multiplier blocks in FIR digital filters," *IEEE Transaction on cicuits Systems II, Analog Digital Signal Processing,* vol. 42, pp. 569-577, 1995.

[69] T. D. Memon, P. Beckett, and Z. M. Hussain, "Analysis and Design of Ternary FIR filter using Sigma Delta Modulation," in *14th IEEE INMIC*, 2009, pp. 476-480.

[70] K. N. Macpherson and R. W. Stewart, "Area Efficient FIR filters for high speed FPGA implementation," *IEE Proc.-Vis. Image Signal Process.,* vol. 153, pp. 711-720, Dec. 2006.

[71] Y. Li, C. Peng, D. Yu, and X. Zhang, "The Implementation methods of High Speed FIR Filter on FPGA," in *Proc. ICSICT'08*, 2008, pp. 2216-2219.
[72] N. Benvenuto, L. E. Franks, and F. S. Hill, "Realization of finite impulse response filters using coefficients +1, 0, and -1," *IEEE Transactions on Communications,* vol. COMM-33, October 1985.
[73] K. Amiri, M. Duarte, J. Cavallaro, C. D. (Xilinx), R. R. (Xilinx), and A. Sabharwal, "FPGA in Wireless Communication Applications," A. Chandrakasan, Ed., ed: Springer, 2010.
[74] C. Siriteanu, S. D. Blostein, and J. Millar, "FPGA-based communications receivers for smart antenna array embedded systems," *EURASIP Journal on Embedded Systems,* vol. 2006, pp. 1-13, 2006.
[75] L. G. Barbero and J. S. Thompson, "FPGA design considerations in the implementation of a fixed-throughput sphere decoder for MIMO systems," in *IEEE International Conference on Field Programmable Logic and Applications (FPL '06)*, Madrid, Spain, 2006, pp. 1-6.
[76] K. Masselos and N. Voros, "Implementation of wireless communications systems on FPGA-based platforms," *EURASIP Journal on Embedded Systems,* vol. 2007, pp. 1-9, 2007.
[77] C. SANDERSON. (October 2007) FPGAs Deliver for Next-Generation Signal Processing Systems. *RTC Magazine*.
[78] C. Dick and F. Harris, "FPGA DSPs-the platform for NG wireless communications," *RF Design,* vol. 23, pp. 56-66, 2000.
[79] M. S. Naghmash, M. F. Ain, and C. Y. Hui, "FPGA implementation of software defined radio model based 16QAM," *European Journal of Scientific Research,* vol. 35, pp. 301-310, 2009.
[80] Enabling Wireless Communication Around the World. Available: http://www.xilinx.com/publications/prod_mktg/wireless_brochure.pdf
[81] T. D. Memon, P. Beckett, and A. Z. Sadik, "Performance-Area Tradeoffs in the Design of Short Word Length FIR Filter," in *5th IEEE ICMENS'09*, 2009, pp. 67-71.
[82] T. D. Memon, P. Beckett, and A. Z. Sadik, "Single-bit and Conventional FIR Filter Comparison in State-of-Art FPGA," in *5th IEEE ICMENS'09*, 2009, pp. 72-76.
[83] T. D. Memon, P. Beckett, and A. Z. Sadik, "Efficient Implementation of Ternary SDM Filters using State-of-the-Art FPGA," *Mehran University Research Journal of Engineering and Technology,* vol. 30, pp. 207-212, 2011.
[84] Tayab D Memon, P. Beckett, and A. Z. Sadik, "Power-Area-Performance Characteristics of FPGA based sigma-delta modulated FIR Filters," *Journal of Signal Processing Systems* 2011 (published online).

[85] K. Wiatr and E. Jamro, "Constant coefficient multiplication in FPGA structures," in *Proceedings of the 26th Euromicro Conference*, 2000, pp. 252-259 vol.1.
[86] A. M. Pervez, H. V. Sorensen, and J. V. D. Spiegel, "An Overview of Sigma-Delta Converters," *IEEE Signal Processing Magazine,* pp. 61-84, January 1996.
[87] C.-W. Ng, N. Wong, and T.-S. Ng, "Bit-Stream Adder and Multiplier for Tri-Level Sigma-Delta Modulators," *IEEE Transaction on Circuits and Systems-II: Express Briefs,* vol. 54, pp. 1082-1086, December 2007.
[88] R. A. Losada, *Digital Filters with MATLAB*: Mathworks Inc 2008.
[89] R. Mehboob, S. A. Khan, and R. Qamar, "FIR Filter Design Methodology for Hardware Optimized Implementation," *IEEE Transaction on Consumer Electronics* vol. 55, pp. 1669-1673, July 2009.
[90] Altera Inc., *Quartus-II Handbook Version 9.1* vol. volume-I: Design and Synthesis: Altera Corporation, 2009.
[91] H. J. M. Veendrick, "Short-Circuit Dissipation of Static CMOS Circuitry and its Impact on the Design of Buffer Circuits," *IEEE Journal of Solid-State Circuits,* vol. 19, pp. 468-473, 1984.
[92] J. Lota, M. Al-Janabi, and I. Kale, "Nonlinear-stability analysis of higher order – modulators for DC and sinusoidal inputs," *IEEE Transactions on Instrumentation and Measurement,* vol. 57, pp. 530-542, 2008.
[93] S. Hein and A. Zakhor, "On the stability of sigma delta modulators," *IEEE Transactions on Signal Processing,* vol. 41, pp. 2322-2348, 1993.
[94] G. K. Ma and F. J. Taylor, "Multiplier policies for digital signal processing," *IEEE ASSP Magazine,* vol. 7, pp. 6-20, 1990.
[95] J. M. de la Rosa, "Sigma-Delta Modulators: Tutorial Overview, Design Guide, and State-of-the-Art Survey," *Circuits and Systems I: Regular Papers, IEEE Transactions on,* vol. 58, pp. 1-21, 2011.
[96] A. C. Thompson, Z. M. Hussain, and P. O'Shea, "A correlative criterion for the stability of sigma-delta based IIR filter: Application to an FIR-like bit-stream filter," in *2nd WSEAS International Conference on Electronics, Control and Signal Processing*, singapore, 2003, pp. 1-4.
[97] T. N. Rajashekhara and I. S. E. Chen, "A fast adder design using signed-digit numbers and ternary logic," in *Proceedings of the 1990 IEEE Southern Tier Technical Conference*, 1990, pp. 187-194.
[98] G. W. Reitwiesner, "Binary arithmetic," *Advances in computers,* vol. 1, pp. 231-308, 1960.
[99] D. Chen, "VHDL Implementation of Fast Adder Tree," Masters, Dept of Electrical Engineering, Linkoping University, 2005.

[100] N. Takagi, H. Yasuura, and S. Yajima, "High-Speed VLSI Multiplication Algorithm with a Redundant Binary Addition Tree," *IEEE Transactions on Computers,* vol. C-34, pp. 789-796, 1985.
[101] A. Z. Sadik and Z. M. Hussain, "Adaptive LMS ternary filtering," 2011, pp. 1-3.
[102] Robert A. Hawley, B. C. Wong, T.-j. Lin, J. Laskowski, and H. Samueli, "Design Techniques for Silicon Compiler Implementations of High-Speed FIR Digital Filters," *IEEE Journal of Solid-State Cicuits,* vol. 31, pp. 656-667, May1996.
[103] Y. C. Lim, J. B. Evans, and B. Liu, "Decomposition of binary integers into signed power-of-two terms," *Circuits and Systems, IEEE Transactions on,* vol. 38, pp. 667-672, 1991.
[104] J. Liu and X. Lin, "Equalization in high-speed communication systems," *IEEE Circuits and Systems Magazine,* vol. 4, pp. 4-17, 2004.
[105] E. Biglieri, J. Proakis, and S. Shamai, "Fading channels: Information-theoretic and communications aspects," *IEEE Transaction on Information Theory,* vol. 44, pp. 2619-2692, 1998.
[106] J. G. Proakis, "Adaptive Equalization for Tdma Digital Mobile Radio," *Ieee Transactions on Vehicular Technology,* vol. 40, pp. 333-341, May 1991.
[107] A. Goldsmith, *Wireless communications*: Cambridge Univ Pr, 2005.
[108] D. Smalley, "Equalization Concepts: A Tutorial," *Atlanta Regional Technology Center, Texas Instruments,* pp. 1-29, 1994.
[109] J. G. Proakis, "Adaptive Equalization Techniques for Acoustic Telemetry Channels," *IEEE Journal of Oceanic Engineering,* vol. 16, pp. 21-31, Jan 1991.
[110] E. Biglieri, J. Proakis, and S. Shamai, "Fading channels: Information-theoretic and communications aspects," *Information Theory, IEEE Transactions on,* vol. 44, pp. 2619-2692, 2002.
[111] J. G. Proakis, *Digital Communication* vol. 4th Edition Mc-Graw-Hill 2004.
[112] V. J. Mathews and S. H. Cho, "Improved Convergence Analysis of Stochastic Gradient Adaptive Filters Using the Sign Algorithm," *Ieee Transactions on Acoustics Speech and Signal Processing,* vol. 35, pp. 450-454, Apr 1987.
[113] T. D. Memon, P. Beckett, A. Z. Sadik, and P.O'Shea, "Single-bit adpative channel equalization for narrowband signals," in *IEEE TENCON*, Bali Indonesia, 22-24 November, 2011.
[114] E. Biglieri, J. Proakis, and S. Shamai, "Fading channels: Information-theoretic and communications aspects," *Ieee Transactions on Information Theory,* vol. 44, pp. 2619-2692, Oct 1998.
[115] A. H. S. Mansour A. Aldajani, "Stability and Performance Analysis of an Adaptive Sigma–Delta Modulator," *IEEE Transactions on Circuits*

and Systems - II: Analog and Digital Signal Processing vol. 48, pp. 233-244, 2001.

[116] G. A. Clark, S. K. Mitra, and S. R. Parker, "Block Implementation of adaptive digital filter," *IEEE Transactions on Circuits and Systems* vol. 29, pp. 744 - 752 1981.

[117] B. Widrow, M. Lehr, F. Beaufays, E. Wan, and M. Bilello, "Learning Algorithms for Adaptive Signal-Processing and Control," *1993 Ieee International Conference on Neural Networks, Vols 1-3,* pp. 1-8, 1993.