EFFICIENT ARCHITECTURAL TRANSFORMATION OF MULTIRATE RECURSIVE FILTERS

Author:

Umar Farooq

01-UET/PhD-EE-01

Thesis Supervisor

Prof. Dr. Habibullah Jamal

A thesis submitted in partial fulfillment of the requirement for the degree of Doctor of Philosophy

Department of Electrical Engineering

University of Engineering and Technology, Taxila, Pakistan

2008

DEDICATION

This thesis is dedicated to my parents and family members for their great love, encouragement, moral and emotional support for all the times.

UNDERTAKING

I certify that research work titled **"Efficient Architectural Transformation of Multirate Recursive Filters"** is my own work. The work has not been presented elsewhere for assessment. Where material has been used from other sources it has been properly acknowledged/referred.

Umar Farooq

01-UET/PhD-EE-01

ACKNOWLEDGMENTS

First of all, I am thankful to Allah Almighty for His countless blessings and special favors in every moment of my life.

I am grateful to my Supervisor, Prof. Dr. Habibullah Jamal, Vice Chancellor, University of Engineering and Technology Taxila for his valuable guidance and continuous encouragement during the entire period of this work. His uninterrupted support greatly assisted me in developing the research skills and achieving the target. I consumed a lot of his personal valuable time for which I am greatly indebted.

I would like to express my profound gratitude to the members of my Research Committee; Prof. Dr. Muhammad Saleem Mian, Chairman EED UET Lahore and Prof. Dr. Muhammad Khawar Islam, UET Taxila, for continuous guidance and motivation. Special thanks are due to Dr. Shoab Ahmed Khan, Centre for Advanced Studies in Engineering (CASE) Islamabad for his kind support and time throughout this work. His timely discussions and creative suggestions are whole heartedly appreciated.

I wish to thank Dr. Stephan Weiss, University of Strathclyde, Glasgow U.K. for his valuable comments and suggestions to improve the thesis.

I owe special thanks to Higher Education Commission and UET Taxila for providing me web connection to Digital Library in my office that helped a lot in the current research. Financial support provided by Ministry of Science & Technology, IT and Telecom Division, Islamabad through the Endowment Fund for research in Signal Processing at UET Taxila, is highly acknowledged.

I thank my colleagues especially from EED and generally from the University who gave me support while doing the research and accomplishing this thesis. The valuable assistance provided by the staff of ASIC Design Lab and Directorate of Advanced Studies, Research and Technological Development (ASRTD) is specially acknowledged.

Table of Contents

INTRODUCTION 14

1.1 Thesis Outline 19

1.2 References 20

CHAPTER-2 29

MERGED DELAY TRANSFORMATION FOR MULTIRATE SIGNAL PROCESSING 29

2.1 Introduction 29

2.2 Merged Delay Transformation 30

2.3 Transformation of Arbitrary Order IIR Filter 38

2.4 Matlab Simulations of Merged Delay Transformation 39

2.5 Conclusions 46

2.6 References 47

CHAPTER- 3 49

EFFICIENT ARCHITECTURES FOR DECIMATION FILTERS 49

3.1 Introduction 49

3.2 Transformation of First Order IIR Filter into an Efficient Decimation Filter 50

3.3 Frequency Response of Transformed Filters 57

3.4 Transformation of Second Order IIR Filter into a Decimation Filter 58

3.5 Frequency Response of Transformed Second Order Filter 62

3.6 Computational Costs 64

3.7 Conclusions 69

3.8 References 70

CHAPTER- 4 72

EFFICIENT ARCHITECTURES FOR INTERPOLATION FILTERS 72

4.1 Introduction 72

4.2 Transformation of First Order IIR Filter into an Efficient Interpolation Filter . 72

4.3 Transformation of Second Order IIR Filter 78

4.4 Frequency Response and Pole-Zero Plots of Transformed Filters 82

4.5 Computational Costs 85

4.6 Conclusions 88

4.7 References 89

CHAPTER- 5 91

HARDWARE IMPLEMENTATIONS AND STABILITY ANALYSIS 91

5.1 Introduction 91

5.2 Hardware Implementation 91

5.3 Effect of Merged Delay Transformation on Stability 94

5.4 Effects of Coefficient Quantization 101

5.5 Conclusions 108

5.6 References 109

CHAPTER- 6 112

CONCLUSIONS 112

APPENDIX-III 122

APPENDIX-IV 125

APPENDIX-V 127

APPENDIX-VI 131

APPENDIX-VII 135

List of Figures

Fig.2. 1 Architecture of 1st order recursive filter after Merged Delay Transformation _____ 32

Fig.2. 2 Simple implementation of Eq.(2.20). _____ 37

Fig.2. 3 Implementation of Nth order IIR Decimation Filter (for N odd). ___ 39

Fig.2. 4 The difference in outputs between original and transformed filters for various values of M _____ 43

Fig.2. 5 Difference between original and transformed output for second order _____ 45

Fig.2. 6 Peak error in the output calculated from real parts only for M = 10. _____ 46

Fig.3. 1 Block diagram of M-to-1 decimator. _____ 50

Fig.3. 2 Noble Identity for down sampler _____ 51

Fig.3. 3 Simple Implementation of Eq(3.3) _____ 52

Fig.3. 4 An M-to-1 down sampler cascaded at the output _____ 54

Fig.3. 5 Invoking Noble Identity and shifting the down sampler _____ 54

Fig.3. 6 Simple identities for multirate interconnected systems _ 55

Fig.3. 7 Shifting down samplers into M-parallel paths _____ 56

Fig.3. 8 Efficient Architecture of First Order IIR Decimation Filter _____ 56

Fig.3. 9 Magnitude and Phase Responses of transformed first order ___ 58

Fig.3. 10 Conversion of IIR Filter into a decimation filter _____ 59

Fig.3. 11 Shifting of down sampler towards left _____ 60

Fig.3. 12 Shifting of down samplers before the M-parallel sub-filters ___ 61

Fig.3. 13 Efficient architecture of second order IIR decimation filter ____ 62

Fig.3. 14 Magnitude and Phase Responses of Original and Transformed Second Order Butterworth Filters (a) M = 3 (b) M = 11. 63

Fig.3. 15 Filter order required for different values of M. 66

Fig.3. 16 Computational Cost for various filter architectures 69

Fig.4. 1 Block diagram of interpolation filter 73

Fig.4. 2 Up sampler shifted after the filter using noble identity. 73

Fig.4. 3 Direct implementation of Eq.(4.3). 74

Fig.4. 41-to L up-sampler introduced at the input. 75

Fig.4. 5 Shifting the up-sampler by applying noble identity. 76

Fig.4. 6 Shifting of 1-to-L up-sampler into L-parallel paths. 77

Fig.4. 7 Efficient Architecture for first order IIR Filter transformed into 78

Fig.4. 8 Simple Implementation of Eq.(4.6) 79

Fig.4. 9 1-to-L up sampler attached at the input side. 80

Fig.4. 10 Shifting right the up-sampler using noble identity 80

Fig.4. 11 Up-sampler shifted to each parallel path. 81

Fig.4. 12 Efficient Architecture of second order IIR filter transformed into interpolation filter 81

Fig.4. 13 Frequency response and pole-zero plots for N = 2, L = 2. 83

Fig.4. 14 Frequency response and pole-zero plots for N = 2, L = 4. 84

Fig.4. 15 Frequency response and pole-zero plots for N = 2, L = 10. 85

Fig.4. 16 Reduction in computational cost 88

Fig. 5.1 Comparison of Power Consumption in mWatt *92*

Fig.5. 2 Comparison of Critical Path Delay in nano seconds 93

Fig.5.3 Hardware Architecture of first order IIR decimation filter 94

Fig.5.4 Pole-zero plots for original and transformed first order filter 96

Fig.5.5 Magnitude and phase response of original and transformed filters (N = 1, L = 8, Butterworth Filter) _____ 97

Fig.5.6 (a) Pole-zero plots for original and transformed filter Section 1 __ 99

Fig.5.7 (a) Magnitude Responses _____ 100

Fig.5.8 The model of a quantizer _____ 102

Fig.5.9 Pole-zero plots for 4 bits, L = 10 _____ 103

Fig.5.10 Pole-zero plots for 6 bits, L = 10. _____ 104

Fig.5.11 Pole-zero plots for 8 bits, L = 10. _____ 105

Fig.5.12 Pole-zero plots for 10 bits, L = 10. _____ 106

Fig.5.13 Pole-zero plots for 8 bits, L = 20. _____ 107

Fig.5.14 Pole-zero plots for 10 bits, L = 20. _____ 107

Fig.5.15 Pole-zero plots for 12-bits L = 20. _____ 108

……..……

List of Tables

Table 3. 1 Filter Specifications. F_s is the input sampling frequency. 64

Table 3. 2 Filter Orders for various implementations from Matlab 65

Table 3. 3 Computational costs of various architectures 68

Table 4. 1 Filter Orders for various implementations from Matlab 86

Table 4. 2 Computational costs of various architectures 87

Table-5.1 Implementation Results *94*

Table- 5. 2 Values of Poles and Zeros for N = 1, L = 8 Butterworth Filter 96

Efficient Architectural Transformation of Multirate Recursive Filters

by

Umar Farooq

ABSTRACT

Computationally efficient architectures of multirate recursive filters are presented in this thesis. An analytical transformation is introduced that converts an IIR filter into an efficient decimation/interpolation filter. The transformation is named as merged delay transformation. This transformation is applicable to first order and second order recursive difference equations. The transfer function of the transformed filter is expressed in the form of $H(z^M)$ so that noble identity of multirate signal processing may be invoked. An N^{th} order filter is required to be implemented in parallel using first order and second order sections.

In case of decimation, a down sampler follows an anti-aliasing filter. With the help of merged delay transformation, the filter is transformed and arranged to provide filtering and down sampling in the same stage. This is possible if the filter is implemented in parallel form. Architecture is introduced where down samplers and delays are arranged on the input side. A commutator switch model operating at an M-times higher rate than the output can replace the input down samplers with successive delays. This results in M-to-1 sample rate reduction without changing the filter characteristics. The frequency response and stability of the filter is not disturbed.

In case of interpolation, an up sampler precedes an anti-imaging filter. Using merged delay transformation we are able to arrange the up samplers after the sub-filters in parallel paths with successive delays. The up samplers and delays are implemented by a commutator switch model operating at L-times faster rate

than the input. Output sampling rate is increased by L and 1-to-L interpolation is achieved. The stability and filter characteristics are unchanged. Filtering and sample rate changes are achieved in the same stage. This avoids the chain of integrators and differentiators as required in a variety of cascade integrator comb (CIC) architectures.

Computational costs in terms of number of multiplies per output sample are compared with polyphase FIR structures and IIR structures. The cost reduction increases with increasing values of M or L. For M = 10, the reduction in cost is 82.64% as compared to FIR decimation filters. As compared to IIR structures, the reduction of the order of 48% is achieved. In case of interpolation, the cost reduction is of the order of 45% as compared to polyphase IIR structures. The reduction in cost is about 68% as compared to polyphase FIR.

The transformed filters are implemented in Verilog HDL and mapped to an FPGA of Spartan-II technology. Parallel implementation of the filters provides benefits of parallel processing. Increased throughput and less hardware requirement are the important characteristics of this architecture. The technique is expected to find wide use in multirate signal processing such as efficient sample rate conversion from CD's to Digital Audio Tape and Digital Transmitter/Receivers.

CHAPTER-1

INTRODUCTION

Multirate filters are digital filters where sampling rate of the input signal is changed into another sampling rate at the output. The sampling rate conversions are required in a variety of modern DSP-based systems [1 – 5]. Several sampling rates are used in digital audio systems, digital transmitters and receivers, fast transforms using filter banks and image processing etc [6 - 11]. Multirate Filters are essentially required when two digital systems with different sampling rates are connected together. Multirate signal processing can greatly increase processing efficiency of DSP-based systems which results in large reductions in system costs [12 – 14].

A multirate system consists of a digital filter and a sample rate changer [15, 16]. The sample rate changer may be a down sampler or an up sampler. In an M-to-1 down sampler, (M - 1) samples are dropped at the output reducing the sample rate by M at the output. The digital filter is a low pass filter or a band pass filter which removes the spectral components of the input signal that may cause aliasing at the lower sampling rate. The digital filter followed by a down sampler is commonly called as a decimator.

An up sampler followed by a digital filter is referred as an interpolator. A 1-to-L up sampler inserts (L – 1) zero valued samples in between two adjacent samples of the input signal. This process produces spectral images of the input signal that have to be removed by an anti-imaging digital filter. When sample rate changes of large value are involved, it is beneficial to carry out overall conversion with multistage system. In this case, filtering and sample rate conversion is required at each stage. Selection of the filter type depends on the design specifications,

clock speed and processing resources. Choice of digital filter has large effect on the computational complexity of a multirate filter [17 – 21].

An important classification of digital filters based on architectural structures corresponds to non-recursive or Finite Impulse Response (FIR) filters and recursive or Infinite Impulse Response (IIR) filters. Usually FIR filters are characterized as linear-phase and stable due to absence of poles other than at zero in the complex z-plane. They are less sensitive to coefficient quantization. The IIR filters are computationally efficient and require less hardware for implementation as compared to FIR filters to achieve the same filter specifications [22]. Both types of filters are used in multirate systems. The use of IIR filters in multirate systems have been limited in the past due to the problems of phase non-linearity, instability and reduced parallelism. Various researchers have done efforts to overcome these limitations of IIR filters [23, 24]. Now, IIR filters can be designed to approximate a linear phase in the pass band or double filtering with block processing technique may be applied for real-time processing [25 – 27]. An IIR decimator or interpolator is highly useful in applications that cannot tolerate very large delays of an FIR decimator or interpolator.

Multirate filtering can be realized by a number of techniques. One of the important class of multirate filters is the half-band filter. These filters have stop band edge frequency exactly at $F_s/4$, where F_s is the sampling frequency. The half-band filters have important characteristics that alternate coefficients of filter impulse response are zero when the filter order is an even number. The odd indexed coefficients are symmetrical so its hardware implementation results in large saving of computational cost. The accurate FIR half-band filter design techniques can be found in [28 – 30]. Half-band IIR filters have fewer multipliers than the FIR filter for the same sharp cutoff specifications. An IIR elliptic half-

band filter is one of the efficient solutions when it is implemented as parallel combination of two all-pass filters [31 – 33]. Single stage half-band filters can provide sample rate conversion of 2 only. However, several stages can be cascaded to achieve sample rate change of any power of 2.

Frequency Response Masking (FRM) approach has been used to realize sharp-transition FIR filters with low complexity in multirate context [34 - 35]. The FRM technique has also been extended to IIR filters to obtain sample rate conversion of power of 2 [36 – 38]. It reduces complexity of computation as compared to direct-form FIR realization but at the cost of increased delay in implementation [39 - 41].

Another well known multirate filtering technique is based on polyphase decomposition. In polyphase decomposition, the filter has to be realized in the direct-form. For sample rate conversion by a factor of M, the filter transfer function is decomposed into M – parallel sub-filters. FIR filters can be easily realized as the parallel combination of polyphase sub-filters. Efficient implementations of decimators and interpolators are possible with the help of polyphase FIR filters [42 – 45]. Polyphase decomposition cannot be directly applied to IIR filter due to the presence of denominator polynomial in its transfer function. However, its numerator can be decomposed into polyphase components similar to FIR filters [46 – 48]. Due to presence of feedback path in its implementation, the applications of polyphase recursive filters in multirate systems have been less explored as compared to FIR filters.

A well-known filter structure for large conversion factors in multirate filters is Cascade Integrator Comb (CIC) filter. The CIC contains two subfilters, the comb and the integrator, that can be arranged in any order [49 – 52]. For up sampling, comb filter is placed at the input of the CIC filter. For down sampling, the comb

filter follows the CIC filter. The pass-band gain of CIC filter is non-uniform which results in distortion of base band spectrum being processed by this filter. A correcting filter embedding inverse response of CIC filter should be attached to get uniform response in the pass band. Moreover stop band attenuation of single 10-tap CIC filter is -13 dB [15]. It is insufficient for most of the applications, so 3-to-5 stages are usually cascaded for reducing the side-lobe level. It can be implemented with the help of delay elements and adders. However, for practical applications, large amount of hardware is required.

Most of the previous work in the area of multirate filters is related with Comb-FIR-IIR filters and polyphase FIR-IIR filters. Naviner et al. [53] suggested the use of Comb-FIR architecture for computationally efficient decimation filters. Aboushady et al. [54] presented an efficient polyphase decomposition of comb filters and provided useful results for the choice of decimation factor in the first stage of multirate multistage systems. Nerurkar et al. [55] implemented a hardware efficient decimation filter by cascading comb filters in the beginning of comb-FIR/IIR chain. Goncalves et al [56] presented efficient decimation filter design based on the polyphase decomposition of IIR filter by multiplying the low pass IIR transfer function by an appropriate polynomial. Russel [22] presented efficient sample rate conversion using IIR filters. Vetterli [57] proposed transform technique for implementation of aperiodic convolution using IIR filters. It requires block processing of IIR filters using block size double the number of poles and equal to power of 2. A variety of multirate filters can be formed using recursive polyphase filters where path transfer functions are all pass recursive filters [15]. However it introduces a large number of pole-zero pairs and the structure is more sensitive to hardware limitations. Direct transformation of recursive filters into efficient multirate filters is not found in the published literature.

In the known multirate filters, multiple architectural stages of the selected filter are needed to attain desired attenuation and other performance characteristics. Filtering and sample rate conversion is performed in a number of different stages. This results in increase in computational costs and hardware requirements. Large reductions in the computational costs are expected if the filter can be directly transformed to give sample rate conversion. An efficient architectural transformation of recursive filters for multirate filtering is greatly desired. This is the main theme of this work.

In this thesis, an efficient architectural transformation is introduced through which sample rate change and filtering can be realized in the same stage. The transformation called as merged delay transformation is applicable to computationally efficient recursive filters. We start with the design of stable IIR filters using the existing filter design techniques. The filter is decomposed into parallel first order sections. Merged Delay Transformation is used to transform the first order recursive filter into a multirate filter. This transformation converts the transfer function of first order IIR filter into the form of $H(z^M)$. Down samplers or up samplers are cascaded to the transformed transfer function. Applying noble identity of multirate signal processing, an efficient architecture of first order recursive decimator/interpolator is obtained. Transformation equations are derived for second order section by combining a pair of first order sections with complex conjugate coefficients. By removing repeated sections in the structure, an optimal architecture is realized for second order section.

Stability of the transformed filter is investigated through pole-zero plots. It is found that the poles of the system always lie inside unit circle in z-plane. In the transformed filter, additional poles and zeros equal in number, are introduced at

the same locations inside the unit circle so magnitude and phase responses are not changed. The parallel implementation has smaller delays and is known to have reduced sensitivity to coefficient quantization [58]. Effect of finite word length in hardware implementation is also explored. The 8-bit implementation shows good results [59]. Several filters are designed and transformed to multirate filters. The magnitude and phase responses of the transformed filters show close agreement with the original filters. Computational cost in terms of number of multiplies per output sample are compared with various conventional structures. The reduction in cost of about 48% and 45% is achieved in decimation and interpolation respectively as compared to conventional IIR structure. Compared with polyphase FIR filters, the reduction in cost is of the order of 82% and 68% for decimators and interpolators respectively. The cost reduction increases with the increase of M or L.

Speed is faster due to parallel implementation of the sub-filters. This architecture requires less hardware and is expected to find wide use in multirate signal processing.

1.1 Thesis Outline

Chapter 2 presents the theoretical background of the merged delay transformation. Here basic concepts of merged delay transformation are introduced. Transformation of first order and second order recursive filters into a multirate filter are explained.

Chapter 3 deals with the transformation of IIR Filter into an efficient decimation filter. It provides the detailed description of architectures for first order, second order and higher order IIR decimators. Various realization structures are explored and an efficient architecture is presented. Computational costs of various

architectures are compared. Frequency responses of the transformed filters are investigated.

Chapter 4 deals with the transformation of IIR Filter into an efficient interpolation filter. The efficient architectures are derived and the computational costs are compared.

Chapter 5 deals with Verilog HDL implementations of IIR multirate filters. Stability of transformed filters is studied with the help of pole-zero plots. Coefficient quantization effects are described for various word length implementations.

Chapter 6 concludes the thesis and gives recommendations for the future work.

1.2 References

[1] Sheikh, F., and Masud, S., "Efficient sample rate conversion for Multi-standard Software Defined Radios," Acoustics, Speech and Signal Processing, 2007, ICASSP 2007. IEEE International Conference on, Volume 2, 15 – 20 April 2007, Page(s): II-329 – II-332.

[2] Crowley R. and Hinman K., "Multirate Signal Processing that enables smart antenna, multi-channel communication, wideband intelligence gathering and data compression systems", Millitary Communication Conference 2005, IEEE MILCOM 2005, 17-20 Oct 2005, Page(s):1026-1032 Vol.2.

[3] Tecpanecatl-Xihuitl, J.L., Kumar, A., and Bayoumi, M.A.**, "**Low complexity decimation filter for multi-standard digital receivers," IEEE International Symposium on Circuits and Systems, ISCAS 2005, Vol.1, 23-26 May 2005, Page(s):552 – 555.

[4] Milic, L.D., and Lutovac, M.D., "Efficient Multirate Filtering using EMQF Subfilters," Telecommunications in Modern Satellite, Cable and

Broadcasting Service, 2003. TELSIKS 2003. 6th International Conference on, Volume-1, 1-3 Oct 2003, Page(s): 301 – 304 vol.1.

[5] Xiqi Gao, Xiaohu You, Bin Sheng, Han Hua, Schulz, E., Weckerle, M., and `Costa, E., "An efficient digital implementation of multicarrier CDMA system based on generalized DFT filter banks," Selected Reas in Communications, IEEE Journal on, Volume 24, Issue 6, June 2006, Page(s): 1189 – 1198.

[6] Mahdi Mottaghi-Kashtiban, Saeed Farazi, and Mahrokh G. Shayesteh, "Optimum Structures for Sample Rate Conversion from CD to DAT and DAT to CD using Multistage Interpolation and Decimation," IEEE International Symposium on Signal Processing and Information Technology, 2006, Page(s): 633-636.

[7] Linjun Xu, and Jun Han, "Cancellation of harmonic interference using multirate signal processing techniques", Instrumentation and Measurement Technology Conference, 2006, IMTC 2006. Proceedings of the IEEE, 24-27 April 2006 Page(s):1392-1396.

[8] Xu, X., Xie, X., and Wang, F., "Digital Up and Down Converter in IEEE 802.16d", Signal Processing, The 8th International Conference on, 2006, Volume 1, Page(s): 16-20.

[9] Wu-Sheng Lu and Ana-Maria Sevcenco, "Design of optimal decimation and interpolation filters for low-bit-rate image coding," APCCAS-2006, IEEE Asia Pacific Conference on Circuits and Systems 2006, Page(s):378-381.

[10] J. L. Tecpanecatl-Xihuitl, R. M. Aguilar-Ponce, M. A. Bayoumi, and B. Zavidovique, "Digital IF decimation filters for 3G systems using pipeline/interleaving architecture," Signal Processing and Its Applications,

2003. Proceedings. Seventh International Symposium on, Volume 2, 1 – 4 July 2003, Page(s): 327 – 330 vol.2.

[11] Wang, Hong, Lu, Youxin, Wang, Xuegang, "Tree structure for channelized digital receivers," Radar, 2006. CIE'06. International Conference on, Oct 2006, Page(s): 1 – 3.

[12] Camino, P., Dallet, D., Quertier, B., Baudry, A., Comoretto, G., and Le Gal, B., "A decimation filter for a very large band signal in radioastronomy," Microelectronics and Electronics Conference, 2007, RME. Ph.D. Research in, 2 – 5 July 2007, Page(s): 265 – 268.

[13] fredric j. harris, and Egg, Benjamin, "Forming narrowband filters at a fixed sample rate with polyphase down and up sampling filters," Digital Signal Processing, 2007, 15th International Conference on, 1 – 4 July 2007, Page(s): 315 – 318.

[14] Mark W. Coffey, "Optimizing Multistage Decimation and Interpolation Processing – Part II," IEEE Signal Processing Letters, Vol.14, No.1, Jan 2007, Page(s): 24-26.

[15] fredric j harris, "Multirate Signal Processing for Communication Systems," Ch-8 - 10, Prentice Hall PTR, New Jersey, 2006.

[16] Alan V. Oppenheim, Ronald W. Schafer with John R. Buck, "Dicrete-Time Signal Processing," Second Edition, Ch. 4, 2000, Pearson Education Asia, India.

[17] M. B. Yeary, W. Zhang, J. Q. Trelewicz, Y. Zhai, and B. Mcguire, "Theory and Implementation of a Computationally Efficient Decimation Filter for Power-Aware Embedded Systems," Instrumentation and Measurement, IEEE Transactions on Volume 55, Issue 5, Oct 2006, Page(s): 1839 – 1849.

[18] M. Yeary, W. Zhang and J. Q. Trelewicz, “A computationally efficient decimation filter design for embedded systems,” Instrumentation and Measurement Technology Conference, 2004. IMTC 04. Proceedings of the 21st IEEE, Volume 2, 18-20 May 2004, Page(s): 913 – 916 Vol.2.

[19] Mark W. Coffey, “Optimizing Multistage Decimation and Interpolation Processing – Part I,” IEEE Signal Processing Letters, Vol.10, No.4, April 2003, Page(s): 107-110.

[20] Abed, K. H., and Nerurkar, S. B., “Low Power and hardware efficient decimation filter”, Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE Volume 1, 16-20 March 2003, Page(s): 454 - 459 Vol.1.

[21] Haddad, F., Bouchakour, R., Rahajandraibe, W., Zaid, L, and Frioui, O., “Radio frequency passive polyphase filter design for wireless communications,” Communication and Information Technologies, 2007, ISCIT’07. International Symposium on, 17 – 19 Oct 2007, Page(s): 264 – 268.

[22] Andrew I. Russell, “Efficient Rational Sampling Rate Alteration using IIR Filters,” IEEE Signal Processing Letters, Volume 7, No.1, January 2000, Page(s): 6 – 7.

[23] Charles M. Rader, “The rise and fall of recursive digital filters,” Signal Processing Magazine, IEEE, Nov. 2006, Page(s): 46 – 49.

[24] Alfonso Fernandez-Vazquez and Gordana Jovanovic-Dolecek, “A New Method for the Design of IIR Filters with Flat Magnitude Response”, IEEE Trans. on Circuits and Systems- I: Regular Papers, vol.53, No.8, August 2006, Page(s):1761-1771.

[25] M.D. Lutovac, and L.D. Milic, “Approximate linear phase multiplierless IIR half-band filter,” IEEE Signal Processing Letters, vol.7, March 2000, Page(s): 52-53.

[26] K. Surma-Aho and T. Saramaki, “A systematic technique for designing approximately linear phase recursive digital filters,” IEEE Transactions on Circuits and Systems-II, vol.46, July 1999, Page(s):956-963.

[27] Shailesh B, Nerurkar and Khalid H. Abed, “Low-Power Decimator Design Using Approximated Linear Phase N-Band IIR Filter,” IEEE Trans. On Signal Processing, Vol.54, No.4, April 2006, Page(s):1550 – 1553.

[28] Saramaki, T., Yli-Kaakinen, J. , “A Novel Systematic Approach for Synthesizing Multiplication-Free Highly-Selective FIR Half-Band Decimators and Interpolators,” Circuits and Systems, 2006, APCCAS 2006, IEEE Asia Pacific Conference on, 4 – 7 Dec 2006, Page(s): 920 – 923.

[29] Gustafsson, Oscar, DeBrunner, Linda S., DeBrunner, Victor, and Johansson, Hakan, “On the design of sparse half-band like FIR filters,” Signals, Systems and Computers, 2007. ACSSC 2007. Conference Record of the Forty-First Asilomar Conference, 4 – 7 Nov 2007, Page(s): 1098 – 1102.

[30] A. N. Wilson, Jr and H. J. Orchard, “A design method for half band FIR filters,” IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications, Volume 45, No.1, Jan 1999, Page(s):95 – 101.

[31] Eren, L., Unal, M., and Devaney, M. J. , “Harmonic Analysis via Wavelet Packet Decomposition using Special Elliptic Half-Band Filters,” Instrumentation and Measurement, IEEE Transactions on, Volume 56, Issue 6, Dec 2007 Page(s): 2289 – 2293.

[32] Millic, L., and Damjanovic, S., “Frequency transformations of half-band Butterworth filters with filter bank applications,” Telecommunications in

Modern Satellite, Cable and Broadcasting Services, 2005. 7th International Conference on, Volume 1, 28 – 30 Sept 2005, Page(s): 107 -110 vol.1.

[33] Elfataoui, M., and Mirchandani, G., "A novel method for generating complex half-band filters," Acoustics, Speech and Signal Processing, 2005. Proceedings (ICASSP'05). IEEE International Conference on, Volume 4, 18 – 23 March 2005, Page(s): iv/381 – iv/384 Vol.4.

[34] Hakan Johansson, "Two Classes of frequency-response masking linear-phase FIR filters for interpolation and decimation," Circuits, Systems, and Signal Processing, Vol.25, No.2, April 2006, Page(s): 175 – 200.

[35] Y. C. Lim and R. Yang, "On the synthesis of very sharp decimators and interpolators using the frequency response masking technique," IEEE Transactions on Signal Processing, Volume SP-53, No.4, April 2005, Page(s): 1387 – 1397.

[36] H. Johansson and L. Wanhammar, "High speed recursive digital filters based on the frequency response masking approach," IEEE Transactions on Circuits and Systems – II, Volume 47, No.1, Jan 2000, Page(s): 48 – 61.

[37] M. D. Lutovac and L. D. Milic, "IIR filters based on frequency-response masking approach," in Proc. 5th Int. Conf. on Telecommunications in Modern Satellite, Cable and Broadcasting Service, Volume 1,part 1, 2001, Page(s): 163 – 170.

[38] O. Gustafsson, H. Johansson and L. Wanhammar, "Single filter frequency masking high-speed recursive digital filters," Circuits, Systems, and Signal Processing, Volume 22, No. 2, Mar./Apr. 2003, Page(s): 219 – 238.

[39] Y. Lian and C. Z. Yang, "Complexity reduction for FRM based FIR filters using the prefilter equalization technique," Circuits, Systems, and Signal Processing, Volume 22, No.2, Mar./Apr. 2003, Page(s): 115 – 135.

[40] T. Saramaki, J. Yli-Kaakinen, and H. Johansson, "Optimization of Frequency-Response Masking based FIR filters," Circuits, Systems, and Computers, Volume 12, No.5, Oct. 2003, Page(s): 563 – 590.

[41] H. Johansson, "Efficient frequency response masking based FIR filter structures for interpolation and decimation," in Proc. Third Int. Workshop on Spectral Methods in Multirate Signal Processing, Barcelona, Spain, Sept. 13 – 14 2003, Page(s): 59 – 62.

[42] Hongjiang Song, "A General Method to VLSI Polyphase Filter Analysis and Design for integrated RF Applications," International SOC Conference, 2006 IEEE, Sept 2006, Page(s): 31 – 34.

[43] Tecpanecatl-Xihuitl, J. Luis, Aguilar-Ponce, Ruth M., Ismail, Yasser, Bayoumi, Magdy A., "Efficient multiplierless polyphase FIR filter based on new distributed arithmetic architecture," Signals, Systems and Computers, 2007, ACSSC 2007. Conference Record of the Forty-First Asilomar Conference, 4 – 7 Nov 2007, Page(s): 958 – 962.

[44] Eghbali, A., Gustafsson, O., Johansson, H., and Lowenborg, P., "On the complexity of multiplierless direct and polyphase FIR structures," Image and Signal Processing and Analysis, 2007, ISPA 2007. 5th International Symposium on, 27 – 29 Sept 2007, Page(s): 200 – 205.

[45] frederic j. harris, C. Dick, and M. Rice, "Digital Receivers and Transmitters using Polyphase Filter Banks for Wireless Communications," IEEE Transactions on Microwave Theory and Techniques, Volume 51, No.4, April 2003, Page(s):1395-1412.

[46] Z. P. Ma and Bosco Leung, "Polyphase IIR Decimation Filter Design for Oversampled A/D Converters with Approximately Linear Phase," IEEE

Transactions on Circuits and Systems –II: Analog and Digital Signal Processing, Volume 39, No.8, August 1992, Page(s): 497 – 505.

[47] Arthur Krukowski and Izzet Kale, "Design of complex polyphase IIR Multi-Flattop filter," ISCAS 2004, International Symposium on Circuits and Systems 2004, Page(s): 553-536.

[48] Mark Alan Sturza, "Differential Evolution Design of Polyphase IIR decimation Filters", Patent Application Publication US 2007/0153946 A1 dated July 5, 2007.

[49] Teymourzadeh, Rozita, Othman, Masuri Bin, "An enhancement of decimation process using fast cascaded integrator comb (CIC) filter," Semiconductor Electronics, 2006, ICSE'06, IEEE International Conference on, Oct 29 2006 - Dec 1 2006, Page(s): 811 – 815.

[50] Wajih A. Abu-Al-Saud and Gordon L. Stuber, "Modified CIC Filter for Sample Rate Conversion in Software Radio Systems", IEEE Signal Processing Letters, vol.10, No.5, May 2003, Page(s):152-154.

[51] Gordana Jovanovic-Dolecek and Sanjit K. Mitra, "Stepped Triangular CIC-Cosine Decimation Filter", IEEE NORSIG 2006, Page(s): 26-30.

[52] Torres, F. J. T., Dolecek, G. J., "Compensated CIC-cosine decimation filter," Communication and Information Technologies, 2007, ISCIT'07. International Symposium on, 17 – 19 Oct 2007, Page(s): 256 – 259.

[53] L. Naviner, J-F Naviner, "On efficient cascade implementation of narrow band decimator filter for SD modulators," Proc. of 43rd IEEE Midwest Symposium on Circuits and Systems, Volume 1, 8 – 11 Aug. 2000, Page(s):70 – 73 vol.1.

[54] H. Aboushady, Y. Dumonteix, M. M. Louerat, and H. Mehrez, "Efficient polyphase decomposition Comb decimation filters in Sigma Delta analog to

digital converter," Transaction IEEE on Circuits and Systems-II, Volume 48, Oct. 2001, Page(s): 898 – 903.

[55] Nerurkar, S. B. and Abed, K. H., "Hardware efficient logarithmic digital decimation filter," Circuits and Systems, 2005, 48th Midwest Symposium on, 7 – 10 Aug. 2005, Volume 1, Page(s): 231 – 234 vol. 1.

[56] Goncalves, M. C., Petraglia, A., "Efficient decimation filter design for lofargram analysis in passive sonar systems," Instrumentation and Measurement, IEEE Transactions on, Volume 55, Issue 1, Feb 2006, Page(s): 132 – 139.

[57] Martin Vetterli, "Running FIR and IIR filtering using multirate filter banks," Acoustics, Speech, and Signal Processing, Trans. IEEE, Volume 36, No.5, May 1988, Page(s): 730 – 738.

[58] Lindberg, Martin and Popp Andreas, "Numerical analysis of finite word length effects in multirate filter system, "Norchip, 2007, 19 – 20 Nov. 2007, Page(s): 1 – 4.

[59] Artur Krukowski, Richard Charles, Spicer Morling and Izzet Kale, "Quantization effects in the polyphase N-path IIR structure," Instrumentation and Measurement, Transactions IEEE on, Volume 51, No.6, December 2002, Page(s): 1271 – 1278.

.....

CHAPTER-2

MERGED DELAY TRANSFORMATION FOR MULTIRATE SIGNAL PROCESSING

2.1 Introduction

Sampling rate change involves the digital blocks of decimation and interpolation [1]. In decimation the output sampling frequency is lower than input sampling frequency. In interpolation it is the opposite. Resampling is sampling rate conversion by a non-integer factor that is achieved by the combination of interpolation and decimation.

Decimation filter includes an anti-aliasing filter followed by down sampler [2, 3]. It is a linear time-variant digital system. In interpolation filter, an up sampler precedes an anti-imaging filter. The digital system may be an FIR system or an IIR system [4, 5, 6]. Both systems have weaknesses and merits [7]. However, IIR systems are well known for their reduced complexity of computation and lower hardware requirement [8]. Polyphase decomposition is an efficient structure for multirate filtering [9, 10], but it is commonly applied to FIR filters. Normally, an IIR filter of much lower order as compared to FIR can meet the design specifications. An IIR filter transfer function has a numerator as well as a denominator polynomial. Polyphase decomposition is not applicable due to the presence of a denominator polynomial. However, an IIR filter can be implemented as cascade of numerator and denominator, and polyphase decomposition can be applied to the numerator only. Such architecture is more efficient as compared to polyphase FIR filters. Filters can be implemented as cascade of second order and first order

sections or parallel sections. Parallel implementations are faster and less sensitive to coefficient quantization. We have explored the possibility of transforming an IIR filter into a multirate filter by implementing it in parallel form. An analytical transformation is described in this chapter.

Merged Delay Transformation of a first order recursive difference equation is introduced in section 2.2.1. The equations are derived analytically starting from the basic difference equation. Section 2.2.2 deals with transformation of second order recursive filters by combining two first order sections in parallel. The case of parallel decomposition when multiple order poles exist is discussed in section 2.2.3. Transformation of arbitrary order IIR filter is described in section 2.3. Numerical simulations of merged delay transformations using Matlab are given in section 2.4. Finally section 2.5 concludes this chapter.

2.2 Merged Delay Transformation

2.2.1 Transformation of First Order Recursive Filters

The input-output relationship for a first order recursive filter can be written as follows.

$$y[n] = py[n-1] + rx[n] \qquad \text{Eq(2.1)}$$

Here x[n] and y[n], represent the input and output samples respectively and p and r are the constants. In a conventional IIR filter operation, the input data rate is identical to the output data rate. In a decimation filter, the output data rate is less than the input data rate so some of the outputs are not required. For example, in a decimator with decimation factor of M, the output should contain

only y[Mn]. From Eq(2.1), y[Mn] cannot be computed until we have computed (M-1) intermediate outputs.

For down sampling by M, we have to compute (M-1) intermediate outputs. This is always the difficulty in recursive systems. In non-recursive systems this is not a problem. In recursive systems, if outputs can be calculated directly from M^{th} old output without computing the intermediate outputs, an efficient architecture is possible. We have introduced a transformation where y[Mn] can be computed directly from single past output. For example, if M = 8 then y[8] can be computed directly from y[0]. We do not need y[1], y[2], y[3], …, y[7]. With this transformation, a first order IIR filter can be converted into a multirate filter. Starting from Eq(2.1), we can write y[n-1] as follows.

$$y[n-1] = py[n-2] + rx[n-1] \qquad \text{Eq(2.2)}$$

Substituting it back to Eq(2.1), we obtain as follows.

$$y[n] = p(py[n-2] + rx[n-1]) + rx[n] \qquad \text{Eq(2.3)}$$

or

$$y[n] = p^2 y[n-2] + prx[n-1] + rx[n]$$

From Eq(2.3), we can compute y[2n] directly from a single previous value. Going one step further and substituting the value of y[n-2] in above equation, we obtain as follows.

$$y[n] = p^3 y[n-3] + p^2 rx[n-2] + prx[n-1] + rx[n] \qquad \text{Eq(2.4)}$$

Repeating (M – 1) similar steps we can write the following general equation.

$$y[n] = p^M y[n-M] + \sum_{k=0}^{M-1} p^k r\, x[n-k] \qquad \text{Eq(2.5)}$$

Eq(2.5) represents the current output y[n] in terms of single previous output y[n-M], the current input x[n] and a number of previous inputs. The first order equation of (2.1) has single pole and zero but it has been transformed to have M number of poles and zeros. This effect has been explored later in the chapter. This equation can be implemented as shown in Fig.2.1.

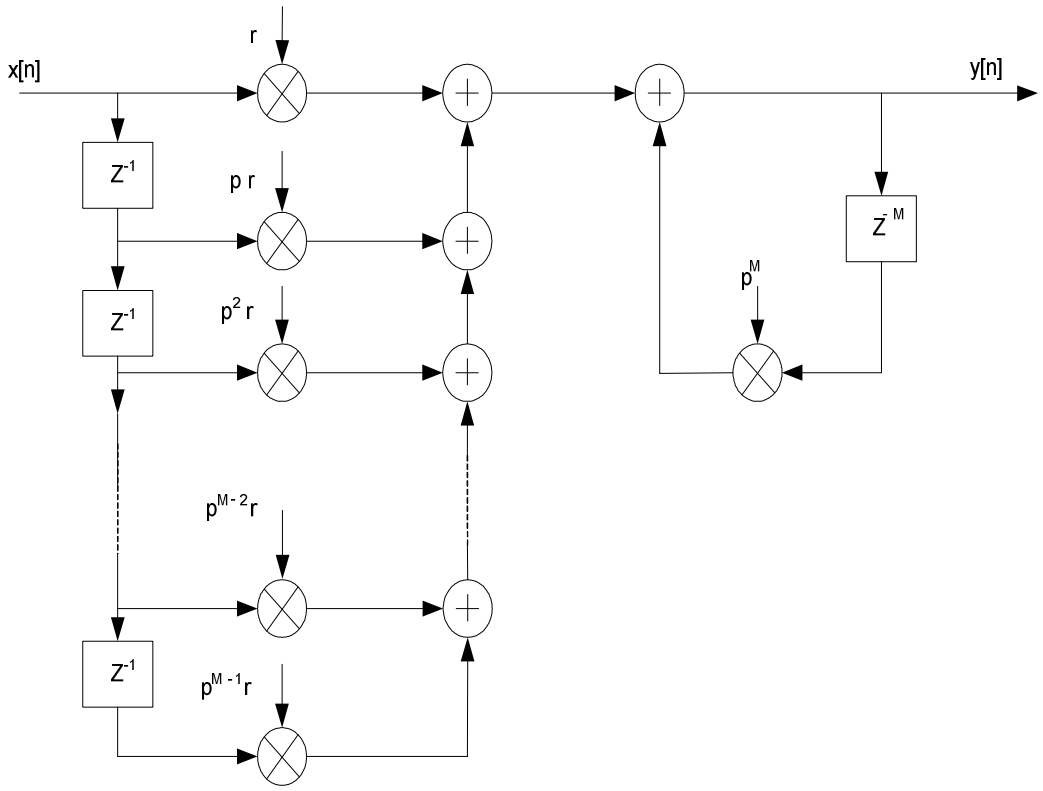


Fig.2. 1 Architecture of 1st order recursive filter after Merged Delay Transformation

The sampling rate change of M is possible in this structure. This structure provides decimated output y[Mn] and M delays at the output side are merged into one delay element operating at a lower rate – the reason for the name "**Merged Delay Transformation".**

Taking z-transform of both sides of Eq(2.5), we obtain the transfer function H(z) as follows.

$$H(z) = \frac{\sum_{k=0}^{M-1} p^k r\, z^{-k}}{1 - p^M z^{-M}} \qquad \text{Eq(2.6)}$$

This is the fundamental relationship that can be implemented for decimation and interpolation. The numerator of above transfer function can be easily partitioned in M parallel paths. Its application for sample rate changes is described in later chapters.

2.2.2 Transformation of Second Order Recursive Filters

The input-output relationship for a simple second order recursive difference equation is written as follow.

$$y[n] = b_0 x[n] + b_1 x[n-1] - a_1 y[n-1] - a_2 y[n-2] \qquad \text{Eq(2.7)}$$

We can write y[n-1] and y[n-2] from Eq(2.7) as follows.

$$y[n-1] = b_0 x[n-1] + b_1 x[n-2] - a_1 y[n-2] - a_2 y[n-3] \qquad \text{Eq(2.8)}$$

$$y[n-2] = b_0 x[n-2] + b_1 x[n-3] - a_1 y[n-3] - a_2 y[n-4] \qquad \text{Eq(2.9)}$$

Substituting the Eq(2.8) and Eq(2.9) in Eq(2.7) we obtain as follows.

$$y[n] = b_0 x[n] + b_1 x[n-1] - a_1 (b_0 x[n-1] + b_1 x[n-2] - a_1 y[n-2] - a_2 y[n-3]) - a_2 (b_0 x[n-2] + b_1 x[n-3] - a_1 y[n-3] - a_2 y[n-4]) \qquad \text{Eq(2.10)}$$

From Eq(2.10) above we see that the computation of output sample y[n] needs all intermediate outputs. We cannot express the current output in terms of single previous output. Due to this reason, all output values have to be computed without skipping any intermediate value that increases the computational costs.

This difficulty is resolved by breaking the transfer function into first order sections. A real coefficient second order system with distinct poles can be decomposed into the first order parallel sections in the following manner. The case of multiplicity of identical poles is discussed later.

$$H(z) = k + H_1(z) + H_2(z) \quad \text{where}$$

$$H_1(z) = \frac{r_1}{1 - p_1 z^{-1}}, \quad H_2(z) = \frac{r_2}{1 - p_2 z^{-1}} \qquad \text{Eq(2.11)}$$

Here k is a real constant and $H_i(z)$ is given as follows.

$$H_i(z) = \frac{r_i}{1 - p_i z^{-1}}, \qquad (i = 1, 2) \qquad \text{Eq(2.12)}$$

In the second order system, the coefficient r_i, p_i may be complex valued. In case of complex valued, the value of r_1 is complex conjugate of r_2. Similarly p_1 and p_2 are complex conjugate. Now, the merged delay transformation equation (2.5) can be applied. We can obtain two outputs $y_1[n]$ and $y_2[n]$ in the following manner.

$$y_1[n] = y_{1R}[n] + j y_{1I}[n] = p_1^M y_1[n - M] + \sum_{k=0}^{M-1} r_1 p_1^k x[n - k] \qquad \text{Eq(2.13)}$$

$$y_2[n] = y_{2R}[n] + jy_{2I}[n] = p_2^M\, y_2[n-M] + \sum_{k=0}^{M-1} r_2\, p_2^k\, x[n-k]$$

Eq(2.14)

The total output $y_{out}[n]$ from a second order section is computed as follows.

$$y_{out}[n] = kx[n] + y_1[n] + y_2[n] \qquad \text{Eq(2.15)}$$

Since r_1, p_1, r_2 and p_2 are complex numbers, so complex multiplications are involved in the implementation of Eq(2.13) and Eq(2.14). Due to complex conjugate nature of r_1 and r_2, p_1 and p_2 we expect some simplification.

It is observed that,

$$y_{1R}[n] = y_{2R}[n]$$

and

$$y_{1I}[n] = -y_{2I}[n]$$

The above results are verified by Matlab simulations for various values of M. The value of $y_{out}[n]$ from a transformed second order IIR filter can be computed as follows.

$$y_{out}[n] = kx[n] + 2y_{1R}[n] \qquad \text{Eq(2.16)}$$

Where

$$y_{1R}[n] = A_{MR}\, y_{1R}[n-M] - A_{MI}\, y_{1I}[n-M] + \sum_{k=0}^{M-1} B_{kR}\, x[n-k] \qquad \text{Eq(2.17)}$$

$$y_{1I}[n] = A_{MI}\, y_{1R}[n-M] + A_{MR}\, y_{1I}[n-M] + \sum_{k=0}^{M-1} B_{kI}\, x[n-k] \qquad \text{Eq(2.18)}$$

The values of A_{MR}, A_{MI}, B_{kR} and B_{kI} are computed as follows.

$$A_{MR} = \text{real}\ (p_1^M)$$
$$A_{MI} = \text{imag}\ (p_1^M)$$
$$B_{kR} = \text{real}\ (p_1^k r_1)$$
$$B_{kI} = \text{imag}\ (p_1^k r_1) \qquad \text{Eq(2.19)}$$

Now, Eq (2.17) & (2.18) for y_{1R} [n] and y_{1I} [n] can be solved simultaneously and the transfer function $H_{1R}(z) = Y_{1R}(z)/X(z)$ can be obtained as follows.

$$H_{1R}(z) = \frac{(1 - A_{MR}z^{-M}) \sum_{k=0}^{M-1} z^{-k} B_{kR} - A_{MI} z^{-M} \sum_{k=0}^{M-1} z^{-k} B_{kI}}{1 - 2A_{MR}z^{-M} + A_{MR}^2 z^{-2M} + A_{MI}^2 z^{-2M}} \qquad \text{Eq(2.20)}$$

Similarly $H_{1I}(z)$, $H_{2R}(z)$ and $H_{2I}(z)$ can be obtained but they are not required. The simplified results are shown above. Its detailed derivation is given at Appendix -1**.** The numerator of $H_{1R}(z)$ can be implemented in M-parallel sub-filters with increasing delays. Each sub-filter should be having two non-zero coefficients; [B_{kR} , (M-1) zeros, - ($A_{MR}B_{kR}$ + $A_{MI}B_{kI}$)] where 0 ≤ k ≤ M - 1. We can express $H_{1R}(z)$ in the following form.

$$H_{1R}(z) = \frac{\sum_{k=0}^{M-1} H_k(z^M) z^{-k}}{1 - A_F z^{-M} - B_F z^{-2M}} \qquad \text{Eq(2.21)}$$

Here we have used the following substitutions.

$$A_F = 2A_{MR}$$

and

$$B_F = -(A_{MR}^2 + A_{MI}^2) \qquad \text{Eq(2.22)}$$

$$H_k(z^M) = B_{kR} - A_{MR}B_{kR}z^{-M} - A_{MI}B_{kI}z^{-M}$$

As the output is real for real values of inputs, Eq (2.21) is implemented to get $y_{1R}[Mn]$ from a second order section.

Simple implementation of Eq.(2.21) is shown in Fig.2.2.

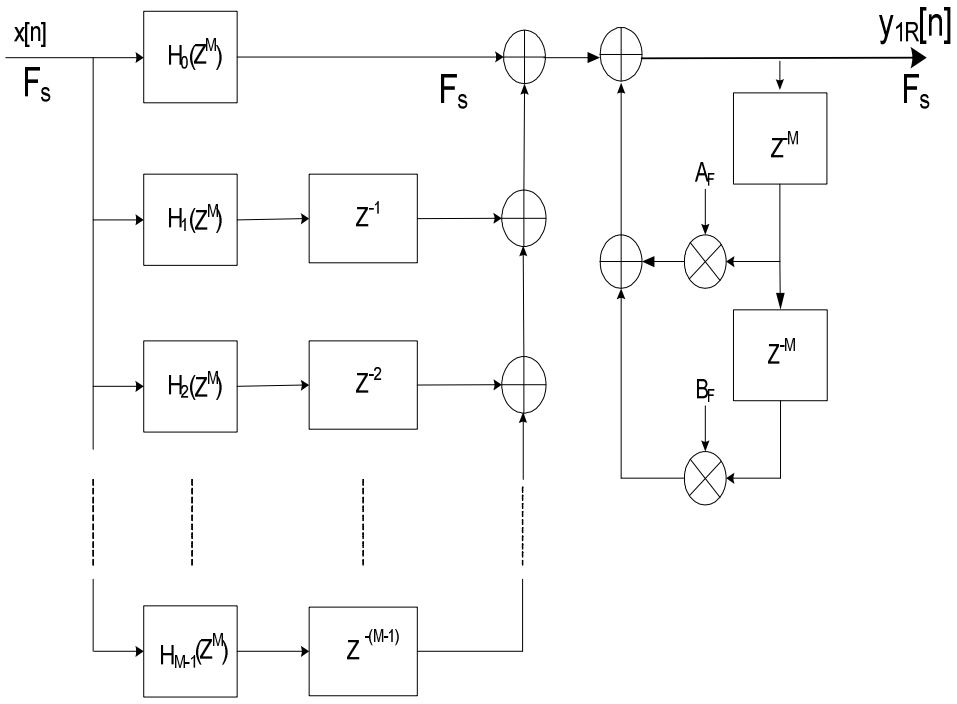


Fig.2. 2 Simple implementation of Eq(2.21).

Computation of one real output in Fig.2.2, is sufficient as the second real output is same. Imaginary outputs are not required in the final result. Implementation of Fig.2.2 is efficient in terms of number of multiplications per output sample as shown later.

2.2.3 Filter Transfer Functions having Multiplicity of Identical Poles

The decomposition introduced in Eq(2.11) is applicable to the filter transfer functions having distinct poles. If more than one pole lie at the same position then simple first order parallel decomposition is not possible. The filter transfer function should be implemented as cascade of first order sections. These first

order sections are then converted to multirate sections using merged delay transformation.

2.3 Transformation of Arbitrary Order IIR Filter

Third and higher order IIR filters cannot be directly transformed using the transformation explained for first and second order filters in Section 2.2. Third and higher order filters are first decomposed into first order parallel sections. Any Nth order IIR filter can be decomposed into N-first order parallel sections in the following manner.

$$H(z) = k + \sum_{i=1}^{N} H_i(z) \qquad \text{Eq(2.23)}$$

$$H_i(z) = \frac{r_i}{1 + p_i z^{-1}} \quad (i = 1, 2, \ldots, N) \qquad \text{Eq(2.24)}$$

Each section of Eq(2.24) can be transformed using merged delay transformation. Outputs from all N section are combined to get the total output.

Although all first order sections can provide the total output but it involves complex multiplications. The computational complexity and hardware requirements are reduced if two first order sections with complex conjugate coefficients are combined to form an optimal second order section. The first order section and second order sections are basic buildings blocks for a higher order filter. An N^{th} order filter can be implemented in the form of N/2 second order sections if N is even. An odd order filter can be implemented as one first order section and (N-1)/2 second order sections.

The filter structure for an N^{th} order IIR filter with N odd, implemented using merged delay transformation is shown in Fig.2.3.

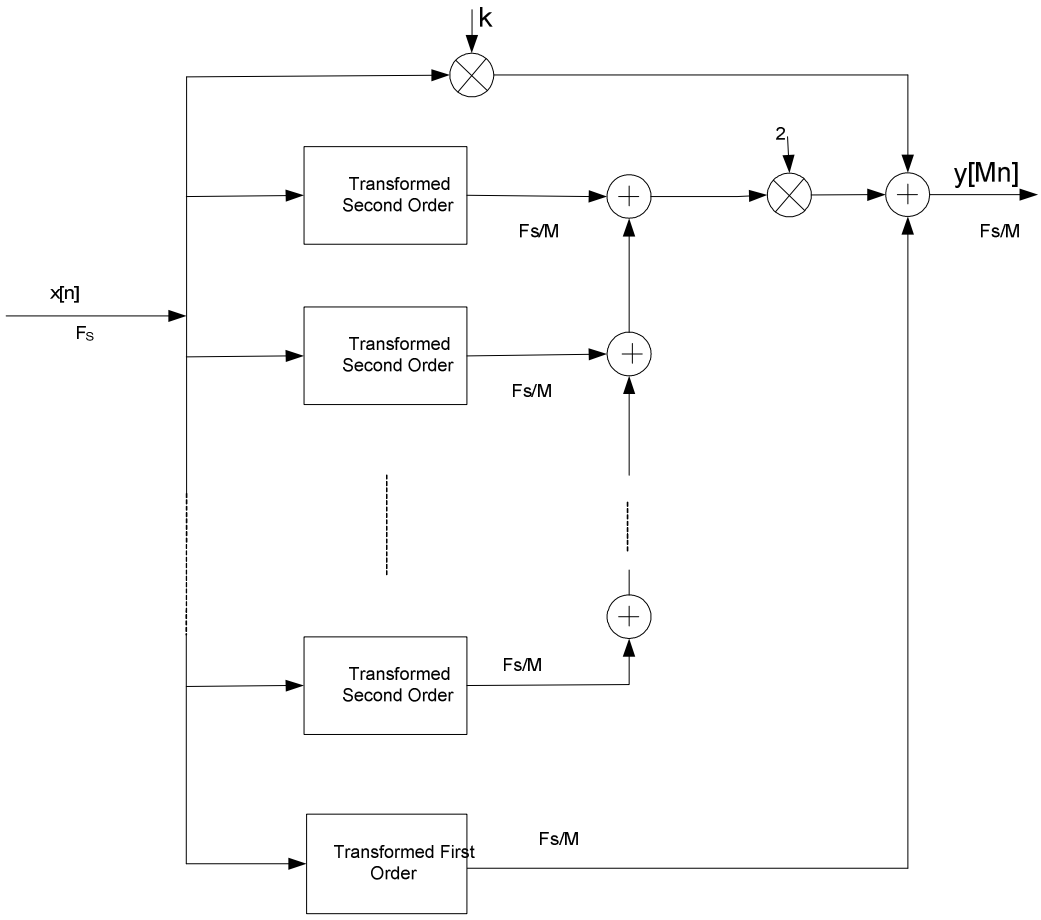


Fig.2. 3 Implementation of Nth order IIR Decimation Filter (for N odd)

2.4 Matlab Simulations of Merged Delay Transformation

2.4.1 Simulation of First Order Recursive Sections Transformed by Merged Delay Transformation

This section describes the numerical simulation of merged delay transformation. The filter design tools of Matlab are used to design an IIR filter of Butterworth, Chebyshev-I, Chebyshev-II and Elliptic types. The output is calculated directly from the original filter equation. We designed an elliptic low pass filter to provide cutoff normalized frequency of 1/M with 1 corresponding to half the sampling rate. It has 0.01 decibels of peak-to-peak ripple in the pass band and a minimum stop band attenuation of 80 decibels. Input signal is generated using the random

values. A Matlab program-A is written that designs the filter for given specifications, calculates the outputs from original filter equation, transforms it into a multirate filter using merged delay transformation and plots the difference between outputs from the original and transformed filter. The M-file of the Program-A is given at **Appendix-II**.

In this program, any filter order may be used. We conducted the simulation by varying the filter order from 1 to 6. Elliptic filter is chosen by using “type=3” in the program. We have different equations for different values of decimation factor M. Any value of M can be used but here it is typically taken as 10. The outputs from the transformed filter equations are again computed with the same inputs. The difference in two outputs is plotted and shown in Fig.2.4.

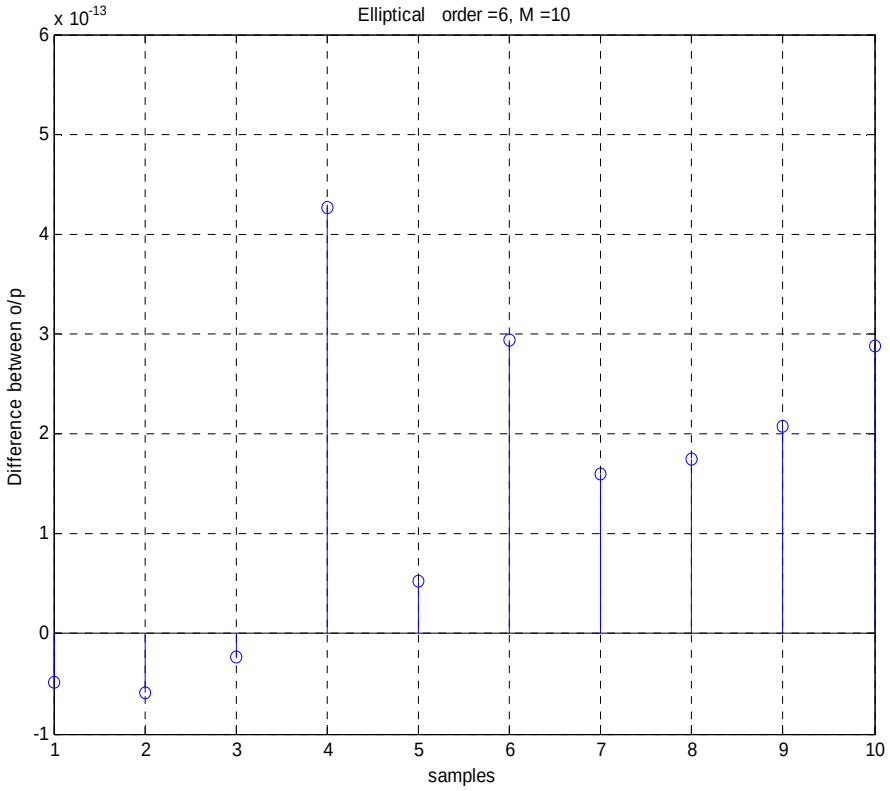
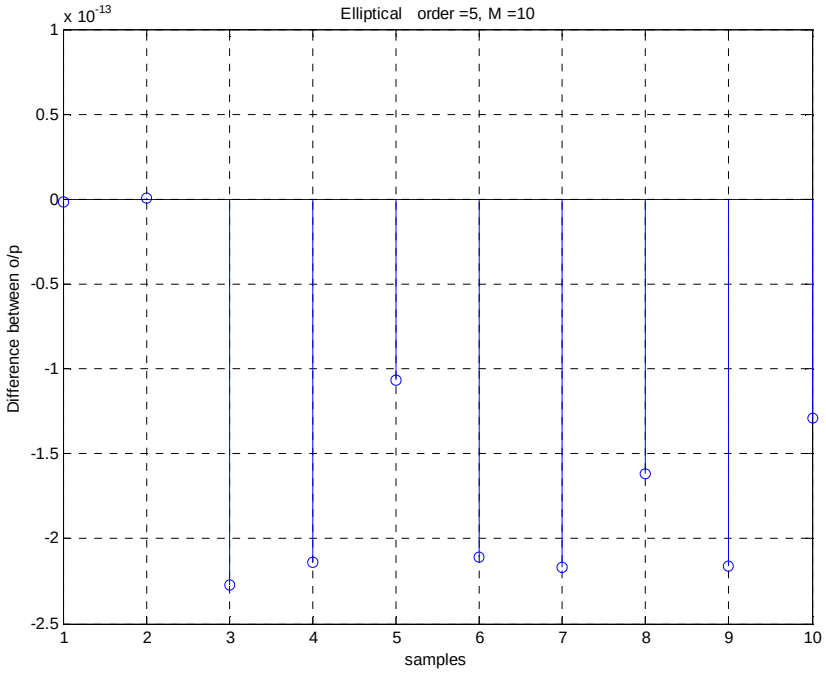


Fig.2.4(a) Filter order is 6, M = 10.



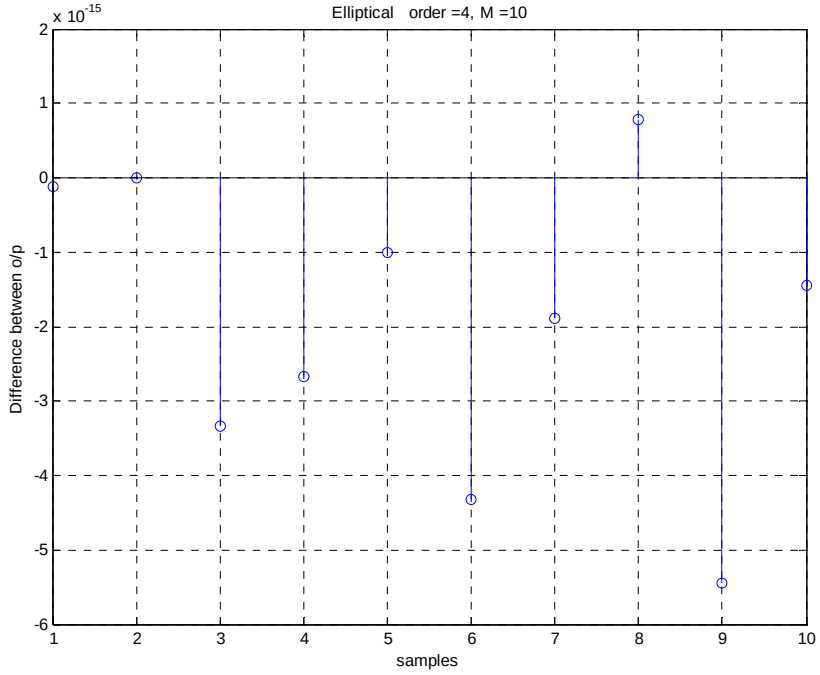
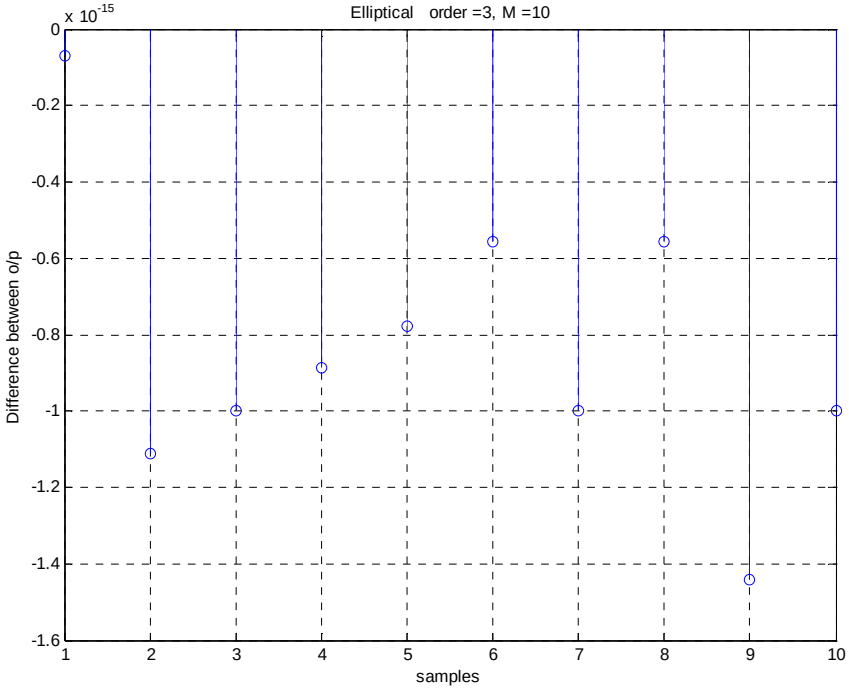


Fig.2.4(b) and (c). Filter order is 5 and 4. M = 10.



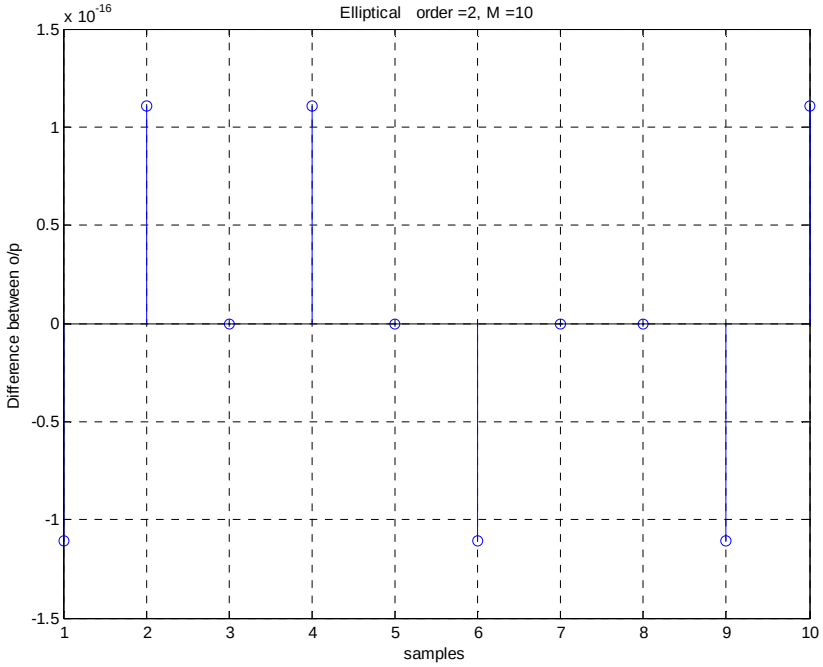


Fig.2.4 (d) and (e) Filter order is 3 and 2. M = 10.

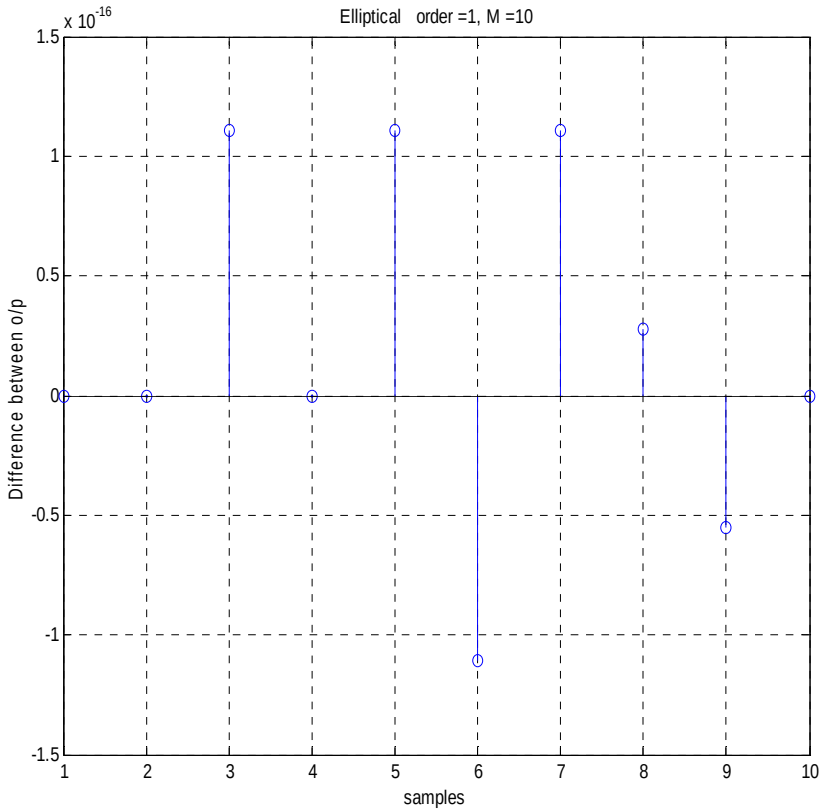


Fig.2.4(f) First order filter, M = 10.

Fig.2.4 The difference in outputs between original and transformed filters for various values of M

Extensive simulations were carried out by changing the type of filter to Butterworth, Chebyshev-I and II, order of filter from 1-10 and changing the value of M from 2 to 10. The input signal is a sequence of random numbers. The difference in output from the original filter and from the transformed filter for various values of M was computed. The simulation results show that the peak error is in the range of 10^{-13} – 10^{-16}. The error increases with the increase in filter order. The lower limit of error is due to Matlab computation. This result shows that the transformed equations are accurately representing the original equations.

2.4.2 Simulations of Second Order Sections by Combining Two First Order Sections with Complex Conjugate Coefficients

We checked the transformation equations for the case of complex coefficients. Since we decompose the higher order filters into first order parallel sections, we obtain pairs of first order sections with complex conjugate coefficients. Due to complex conjugate coefficients, separate outputs for real and imaginary parts are to be computed. From Matlab simulations it was observed that the real parts of output from two first order sections with complex conjugate coefficients are always equal. The imaginary parts of outputs are always equal and opposite. This was verified by running the program B given at **Appendix-III**. It was observed that the calculated values of y_{1R} and y_{2R} are same. The calculated values of y_{1I} are negative of y_{2I}.

The second order section comprising of two first order parallel sections with complex conjugate coefficients was investigated through Matlab simulations. Equations of transformed filter were derived and shown in previous sections. The optimal structure for reduced complexity second order section was simulated and compared with the original filter. The outputs from the original and transformed filter were obtained with the same input samples. The results for M = 4 are shown in Fig.2.5. The program C given at **Appendix-IV** was used for simulation.

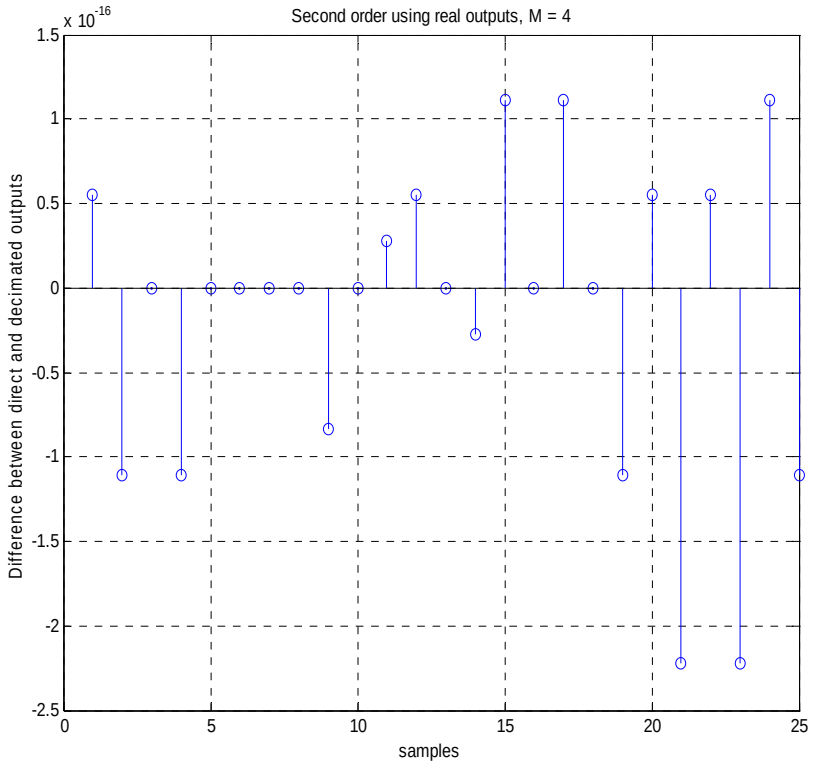
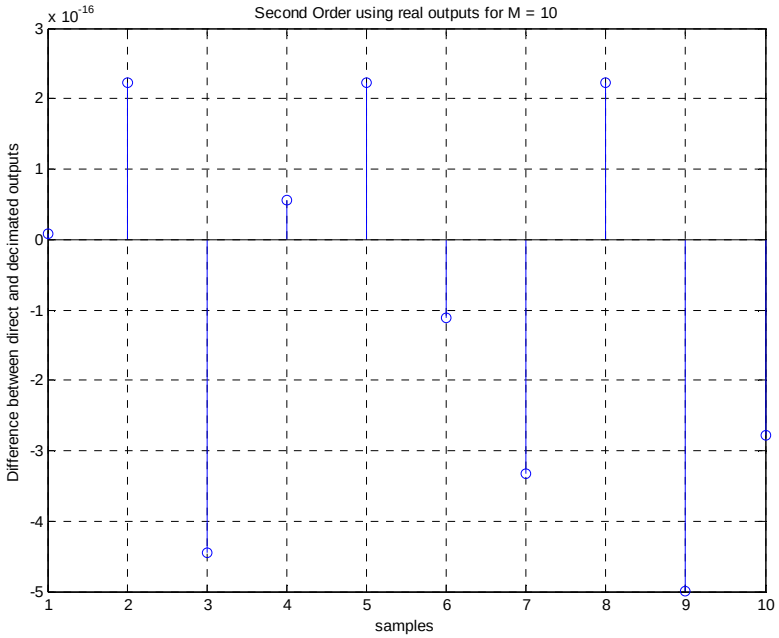


Fig.2. 5 Difference between original and transformed output for second order section, M = 4

The results for M =10 are shown in Fig.2.6. The program D given at **Appendix-V** was used for simulation. Peak error in both the cases is in the range of 10^{-16}. Similar results are obtained for other values of M. This verifies the correctness of transformed equations.



.

Fig.2. 6 Peak error in the output calculated from real parts only for M = 10.

2.5 Conclusions

Merged Delay Transformation was applied to first order recursive filter. Relevant equations were derived and simple structure was presented. Second order recursive sections were decomposed into first order parallel sections with complex conjugate coefficients. Merged delay transformation was applied to second order sections and an efficient architecture was introduced. These structures will be used in subsequent chapters for decimation/interpolation. Matlab simulations were conducted to check the correctness of transformation. The difference between original output and transformed output was found to be negligible.

2.6 References

[1] fredric j harris, “Multirate Signal Processing for Communication Systems,” Ch-8, Prentice Hall PTR, New Jersey, 2006.

[2] Tecpanecatl-Xihuitl, J.L., Kumar, A., and Bayoumi, M.A.**, “**Low complexity decimation filter for multi-standard digital receivers,” IEEE International Symposium on Circuits and Systems, ISCAS 2005, Vol.1, Page(s):552 – 555, 23-26 May 2005.

[3] M. B. Yeary, W. Zhang, J. Q. Trelewicz, Y. Zhai, and B. Mcguire, “Theory and Implementation of a Computationally Efficient Decimation Filter for Power-Aware Embedded Systems,” Instrumentation and Measurement, IEEE Transactions on Volume 55, Issue 5, Oct 2006, Page(s): 1839 – 1849.

[4] Hakan Johansson, “Multirate IIR Filter Structures for Arbitrary Bandwidths”, IEEE Trans. on Circuits and Systems-I:Fundamental Theory and Applications, vol.50, No.12, Dec 2003, pp.1515-1529.

[5] Shailesh B, Nerurkar and Khalid H. Abed, “Low-Power Decimator Design Using Approximated Linear Phase N-Band IIR Filter,” IEEE Trans. On Signal Processing, Vol.54, No.4, Page(s):1550 – 1553, April 2006.

[6] Sheikh, F., and Masud, S., “Efficient sample rate conversion for Multi-standard Software Defined Radios,” Acoustics, Speech and Signal Processing, 2007, ICASSP 2007. IEEE International Conference on, Volume 2, 15 – 20 April 2007, Page(s): II-329 – II-332.

[7] Ljiljana Milic, Tapio Saramaki and Robert Bregovic, “Multirate Filters: An Overview,” APCCAS 2006, IEEE, pp.912-915.

[8] Shailesh B, Nerurkar and Khalid H. Abed, “Low-Power Decimator Design Using Approximated Linear Phase N-Band IIR Filter,” IEEE Trans. On Signal Processing, Vol.54, No.4, Page(s):1550 – 1553, April 2006.

[9] Mark Alan Sturza, “Differential Evolution Design of Polyphase IIR decimation Filters”, Patent Application Publication US 2007/0153946 A1 dated July 5, 2007.

[10] frederic j. harris, C. Dick, and M. Rice, “Digital Receivers and Transmitters using Polyphase Filter Banks for Wireless Communications,” IEEE Transactions on Microwave Theory and Techniques, vol.51, No.4, pp.1395-1412, April 2003.

……

CHAPTER- 3

EFFICIENT ARCHITECTURES FOR DECIMATION FILTERS

3.1 Introduction

Conventional IIR filter has same sampling rate at the input and the output. In a multirate filter, sampling rates are different. To implement decimation, all outputs are computed and some of them are dropped after computation. Efficient architectures [1, 2] are possible if only desired outputs are computed without calculating the intermediate outputs that are to be dropped. Merged delay transformation is applied to an IIR filter and efficient architectures for decimation filters using IIR filter are extracted.

Transformation of first order IIR filter into an efficient decimation filter is described in section 3.2. Relevant equations are derived and architectures are presented. Frequency response of the transformed filter is simulated in section 3.3. Section 3.4 explains the transformation of second order IIR filter into an efficient decimation filter. Section 3.5 provides simulation results for transformed second order filters. Computational costs of various architectures are compared in section 3.6. Section 3.7 concludes the chapter.

3.2 Transformation of First Order IIR Filter into an Efficient Decimation Filter

The equation (2.5) derived in section 2.2 is implemented to transform an IIR filter into a decimation filter. A block diagram showing the process of decimating a signal x[n] by an integer factor M is shown in Figure 3.1 [3, 4].

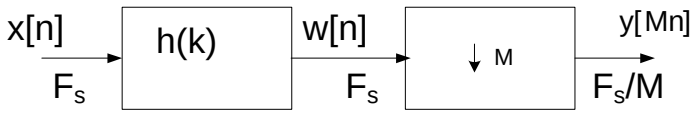


Fig.3. 1 Block diagram of M-to-1 decimator

It consists of a anti-aliasing filter h(k), followed by a down sampler, represented by a down-arrow and an integer factor M. The down sampler reduces the sampling frequency from F_s to F_s/M. To prevent aliasing at the lower rate the digital low pass or band pass filter is used to band limit the input signal to less than $F_s/2M$. Sampling rate reduction is achieved by discarding (M – 1) samples for every M samples of the filtered signal, w(n). The input-output relationship for the decimation process is shown below.

$$y[n] = w(nM) = \sum_{k=-\infty}^{\infty} h(k)x(nM-k) \qquad \text{Eq(3.1)}$$

where

$$w(n) = \sum_{k=-\infty}^{\infty} h(k)x(n-k) \qquad \text{Eq(3.2)}$$

We can interchange the operations of filters and down-sampler/up-sampler with

the operations of down-sampler/up-sampler and filter with appropriate changes. The process that accomplishes this interchange is known as Noble Identity [5]. According to this identity, the filter processing every M^{th} input sample followed by an M-to-1 down sampler gives the same output as an input M-to-1 down sampler followed by a filter processing every M^{th} input sample. The Noble Identity is compactly presented in Fig.3.2 [6, 7, and 8].

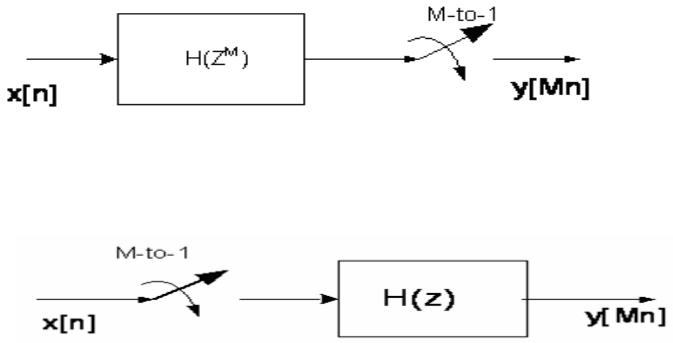


Fig.3. 2 Noble Identity for down sampler

From the figure, we see that the output from a filter $H(z^M)$ followed by an M-to-1 down sampler is identical to an M-to-1 down sampler followed by the filter H(z). The z^M in the filter transfer function tells us that the coefficients are separated by M-samples rather than the one sample delay between coefficients in the filter H(z). It is based on the fact that the output from a M-to-1 down sampler that is obtained from a filter which processes every M^{th} input sample is the same as the output where input is first passed through a M-to-1 down sampler and then operating the filter on reduced samples. This works because M-samples delay at

the input sampling rate is the same interval as one-sample at the output sampling rate. This fact is very useful for efficient implementations.

If we are able to convert H(z) into $H'(z^M)$ then noble identity of multirate signal processing can be applied and the down sampling and filtering may be combined. H(z) for the transformed first order IIR filter was derived in section 2.2 and is reproduced as follows.

$$H(z) = \frac{\sum_{k=0}^{M-1} p^k r\, z^{-k}}{1 - p^M\, z^{-M}}$$

The above transfer function can be easily partitioned in M parallel paths as shown in Fig. 3.3 [9, 10].

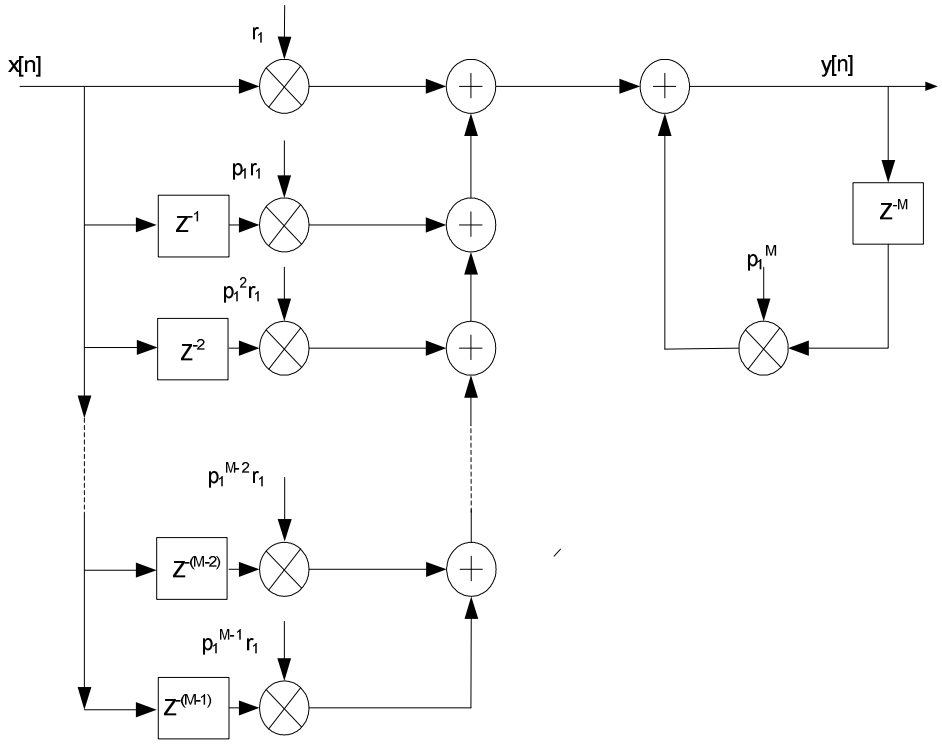


Fig.3. 3 Simple Implementation of Eq(3.3)

The sampling rate is same at the input and output side of Fig.3.3. The transformed form of IIR equation is suitable for conversion into a down sampling filter. The transfer function H(z) is arranged in the following manner.

$$H(z) = \frac{\sum_{k=0}^{M-1} H_k(z^M)\, z^{-k}}{1 - p^M z^{-M}} \qquad \text{Eq(3.3)}$$

where

$$H_k(z) = \sum_{n=-\infty}^{\infty} h_k[n] z^{-n}$$

with

$$h_k[n] = r p^{k+nM} \qquad 0 \le k \le M-1$$

We can arrange H(z) in the form of $H'(z^M)$ where noble identities of multirate signal processing can be applied. Thus,

$$H(z) = H'(z^M) = \frac{\sum_{k=0}^{M-1} H_k(z^M)\, z^{-k}}{1 - p^M z^{-M}} \qquad \text{Eq(3.4)}$$

The numerator is implemented into M parallel paths. To reduce the sampling frequency from F_s to F_s/M, an M-to-1 down sampler is connected at the output as shown in Fig.3.4.

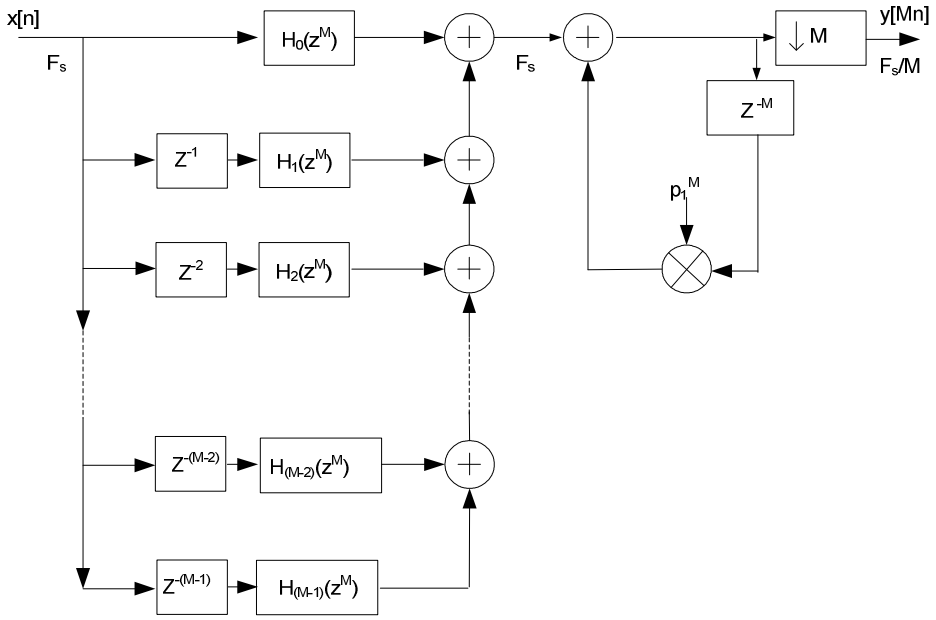


Fig.3. 4 An M-to-1 down sampler cascaded at the output

Using noble identity, the down sampler is shifted towards the left of recursive part as shown in Fig.3.5.

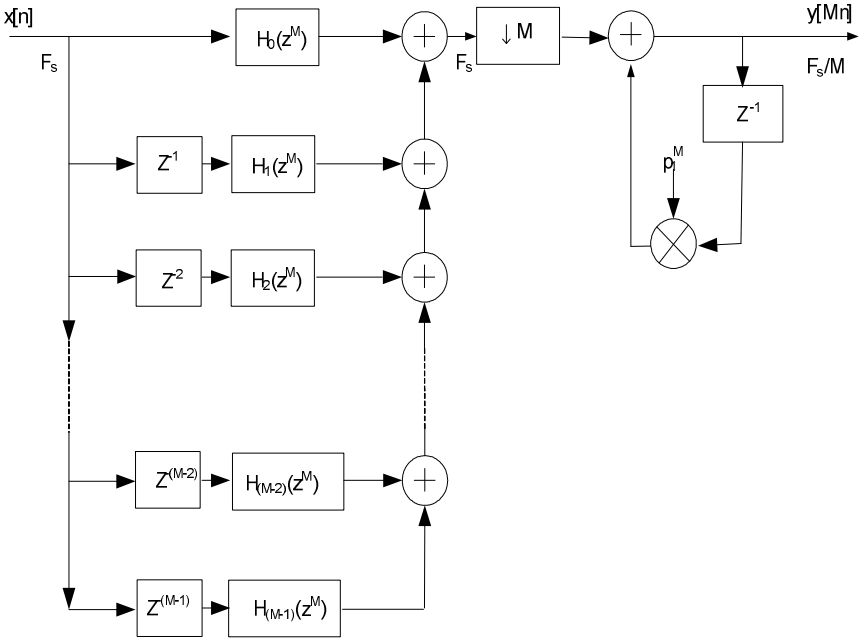


Fig.3. 5 Invoking Noble Identity and shifting the down sampler

In Fig.3.5, the M-number of delays (Z^{-M}) in the recursive part are merged into single delay (Z^{-1}) operating at slower clock F_s/M.

For multirate interconnected systems, the following identity holds. Down samplers can be transferred as shown in Fig.3.6.

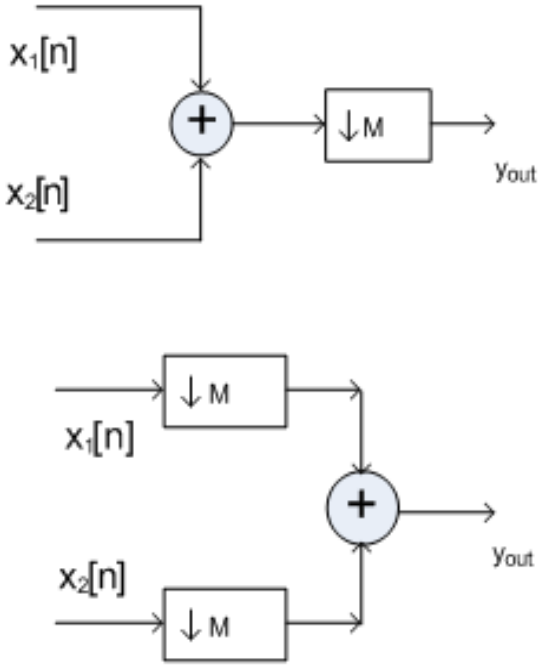


Fig.3. 6 Simple identities for multirate interconnected systems

Using this identity in Fig.3.5, the down sampler is shifted into M-parallel paths with increasing delays. In this way, the structure of Fig.3.7 is obtained.

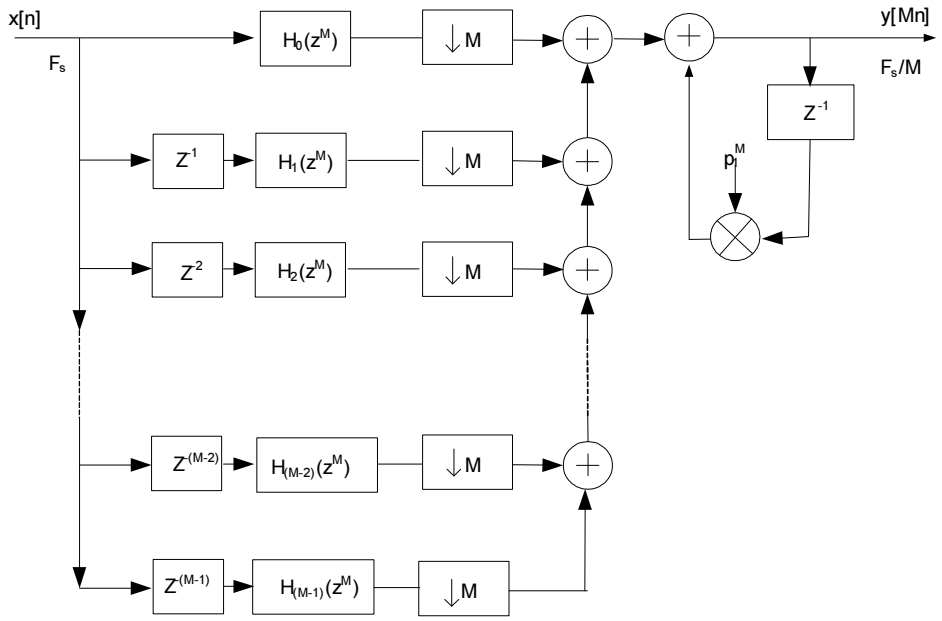


Fig.3. 7 Shifting down samplers into M-parallel paths

In Fig.3.7, the down samplers associated with increasing delay elements can be implemented by a commutator switch model moving in counter-clockwise direction at the input side. This results in sample rate reduction or decimation. The efficient architecture for transformed filter is finally obtained as shown in Fig.3.8.

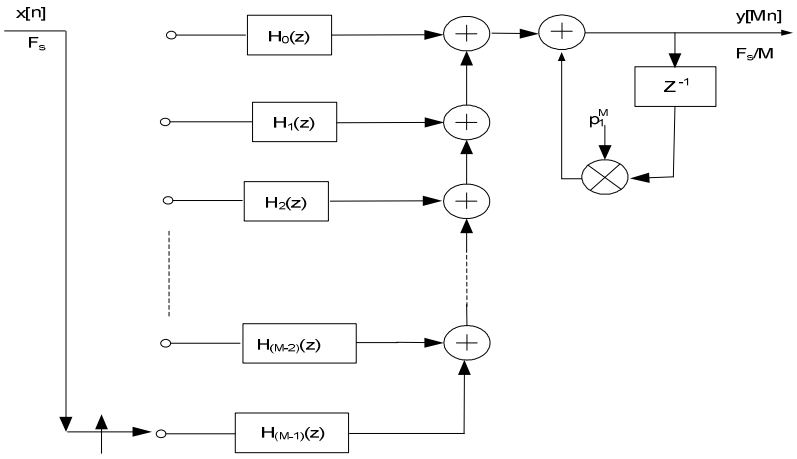


Fig.3. 8 Efficient Architecture of First Order IIR Decimation Filter

3.3 Frequency Response of Transformed Filters

The magnitude and phase responses of the original and transformed filters were simulated. At first, first order sections were studied for different values of decimation factor M. The results are shown in Fig.3.9.

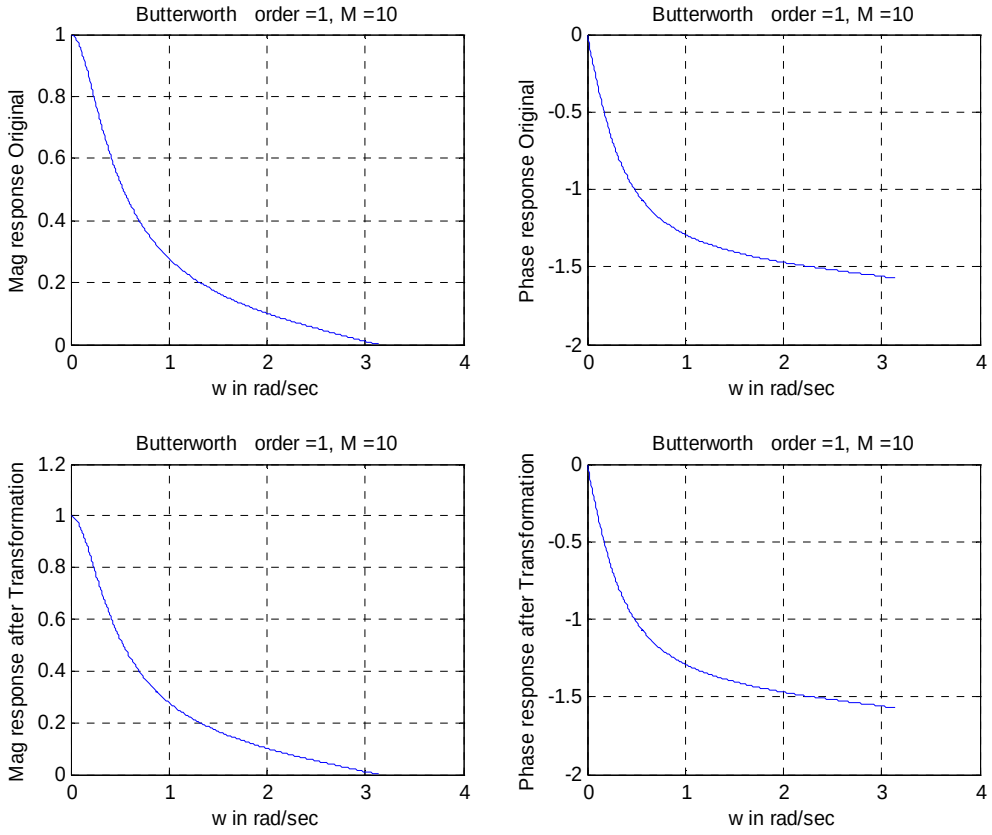


Fig.3.9(a)

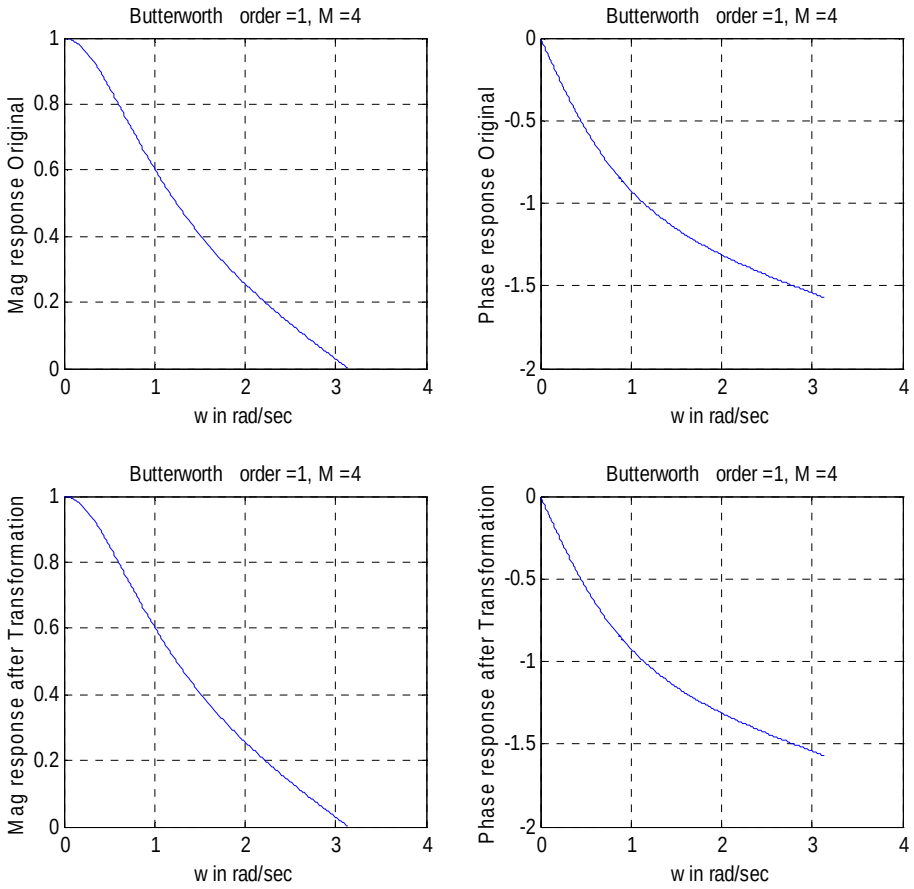


Fig.3.9(b)

Fig.3. 9 Magnitude and Phase Responses of transformed first order IIR Filters (a) M = 10, (b) M = 4.

The magnitude and phase responses are not changed by this transformation. This is very useful result. Due to this reason, filtering and down sampling is achieved in the same stage. The Program-E used for this simulation is shown at **Appendix-VI**.

3.4 Transformation of Second Order IIR Filter into a Decimation Filter

The transfer function $H_{1R}(z) = Y_{1R}(z)/X(z)$ was derived at Eq.(2.20) and is reproduced below.

$$H_{1R}(z)=\frac{(1-A_{MR}z^{-M})\sum_{k=0}^{M-1}z^{-k}B_{kR}-A_{MI}z^{-M}\sum_{k=0}^{M-1}z^{-k}B_{kI}}{1-2A_{MR}z^{-M}+A_{MR}^{2}z^{-2M}+A_{MI}^{2}z^{-2M}}$$

The implementation of above function was shown in Fig.2.2. Sampling rate is same at the input and output side of Fig.2.2. We can convert this structure into an efficient down sampling filter.

To get sample reduction by a factor M, we attach an M-to-1 down sampler after the filter as shown in Fig.3.10.

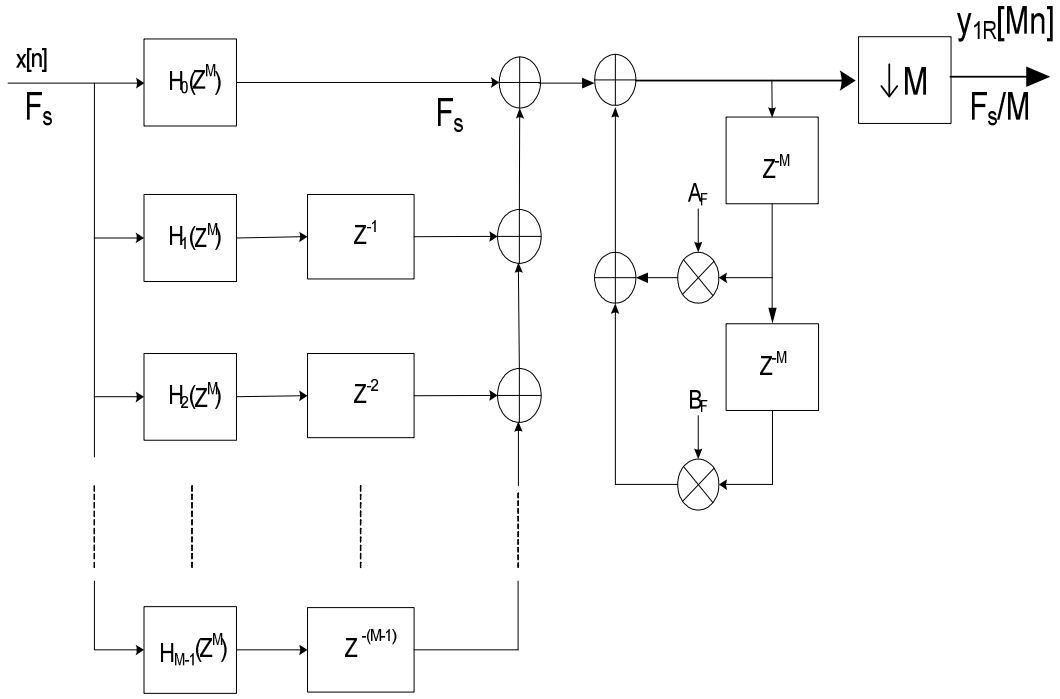


Fig.3. 10 Conversion of IIR Filter into a decimation filter

Using simple identities of multirate interconnected systems, M-to-1 down sampler can be shifted to the left side of recursive part. Using noble identities, z^{-M} in the recursive part are changed into z^{-1}. The transformed structure becomes as shown in Fig.3.11.

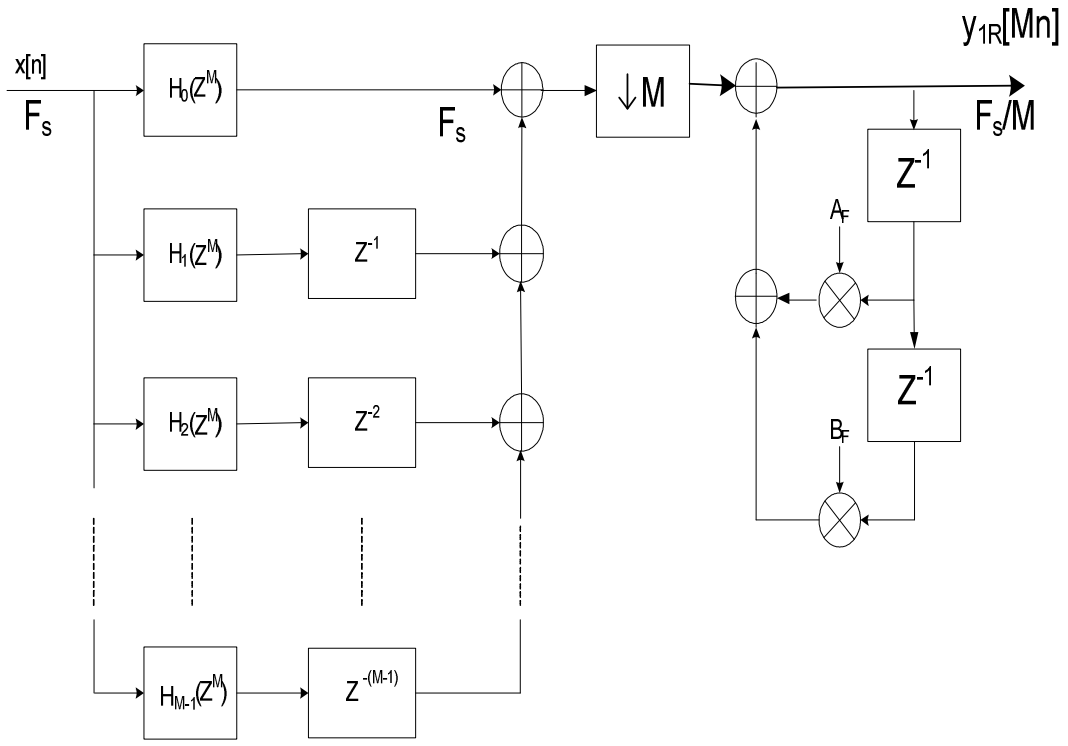


Fig.3. 11 Shifting of down sampler towards left

Again simple identities of multirate systems are applied and M-to-1 down sampler is shifted into each of the M-paths of Fig.3.11. Moreover, it can be placed before the sub-filters using noble identity. In this way, Fig.3.11 is transformed to Fig.3.12.

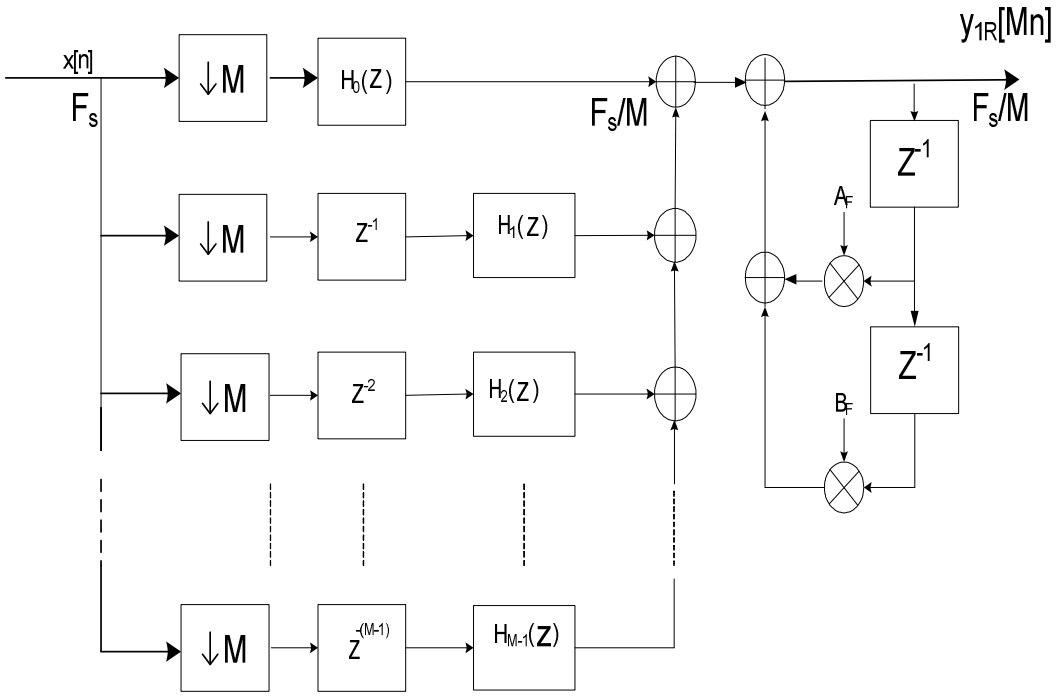


Fig.3. 12 Shifting of down samplers before the M-parallel sub-filters

Down samplers along with increasing delays are implemented by a commutator switch model moving in counter clock wise direction. Down samplers and increasing delays are replaced and the resulting architecture is shown in Fig.3.13.

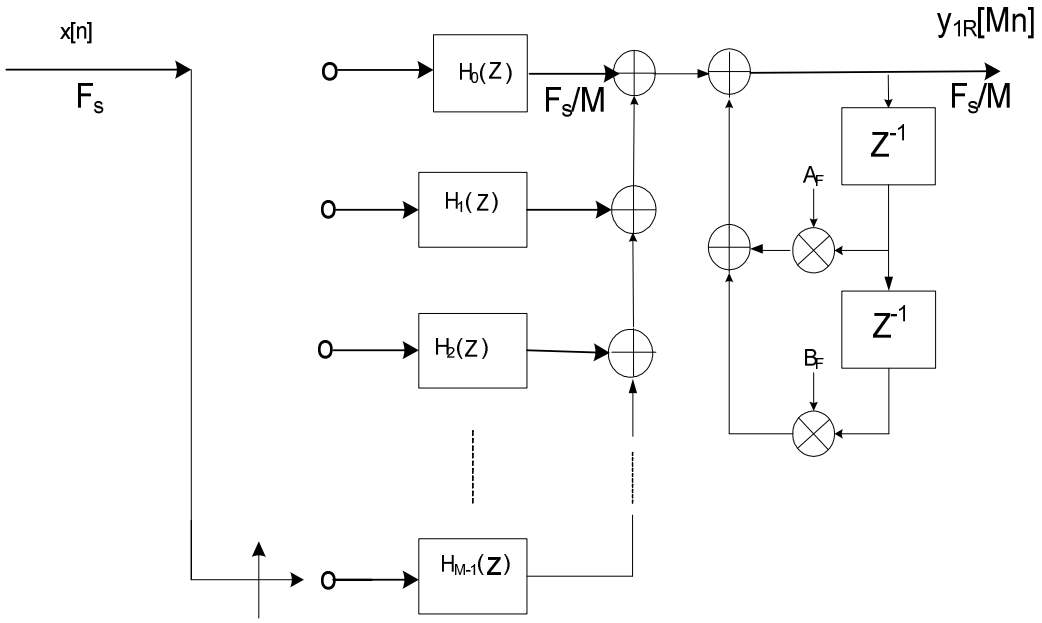


Fig.3. 13 Efficient architecture of second order IIR decimation filter

Implementation of Eq(2.22) resulted in efficient architecture of Fig.3.13. Computation of one real output is sufficient as the second real output is same. Imaginary outputs are not required in the final result.

3.5 Frequency Response of Transformed Second Order Filter

Using the same program-E, the magnitude and phase responses of a second order filter were investigated for various values of M. Typical results for M = 3 and M = 11, are shown in Fig.3.14. From the figures, we find that the frequency response of transformed second order filter is same as of the original filter. The decimation and filtering is achieved in the same stage.

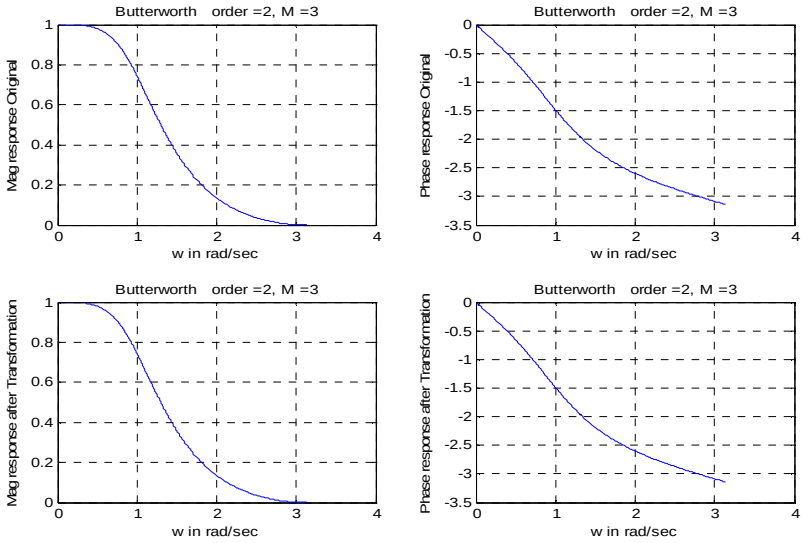


Fig.3.14(a)

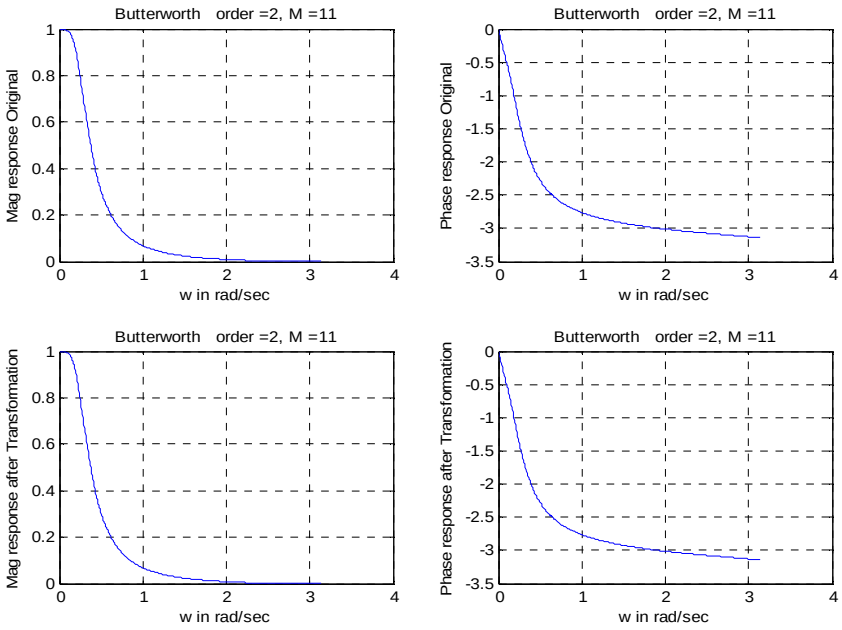


Fig.3.14(b)

Fig.3. 14 Magnitude and Phase Responses of Original and Transformed Second Order Butterworth Filters (a) M = 3 (b) M = 11.

3.6 Computational Costs

In this section we compare the computational costs of various implementations of decimation filters, measured in multiplications per output sample. We use the filter specifications shown in Table 3.1.

Table 3.1 Filter Specifications. F_s is the input sampling frequency.

Specification	Frequency Band	Value (dB)
Pass-Band Ripple	< 0.45 F_s	0.1
Stop-Band Attenuation	>0.55 F_s	50

In case of decimation, where low pass filter is followed by an M-to-1 down sampler, the filter's frequency edges must satisfy the following relationships.

$$f_{passband} = \frac{0.45F_s}{M}$$

and

$$f_{stopband} = \frac{0.55F_s}{M} \qquad \text{Eq(3.5)}$$

We consider the implementation of this decimation filter using conventional IIR filters, transformed IIR filters and polyphase FIR filters. Each of these filters is designed to meet the specifications of Table 3.1. The pass-band and stop-band frequencies were calculated for various values of M, using Eq.(3.5). The minimum required filter orders obtained by Matlab are shown in Table 3.2.

Table 3.2 Filter Orders for various implementations from Matlab

Filter Type	Filter order								
M =	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
FIR Equiripple	48	72	97	121	145	169	193	217	242
IIR Butterworth	25	32	35	36	37	37	38	38	38
IIR Chebyshev	10	12	13	13	13	13	13	13	13
IIR Elliptical	6	7	7	7	7	7	7	7	7

The data of Table 3.2 is plotted in Fig.3.15.

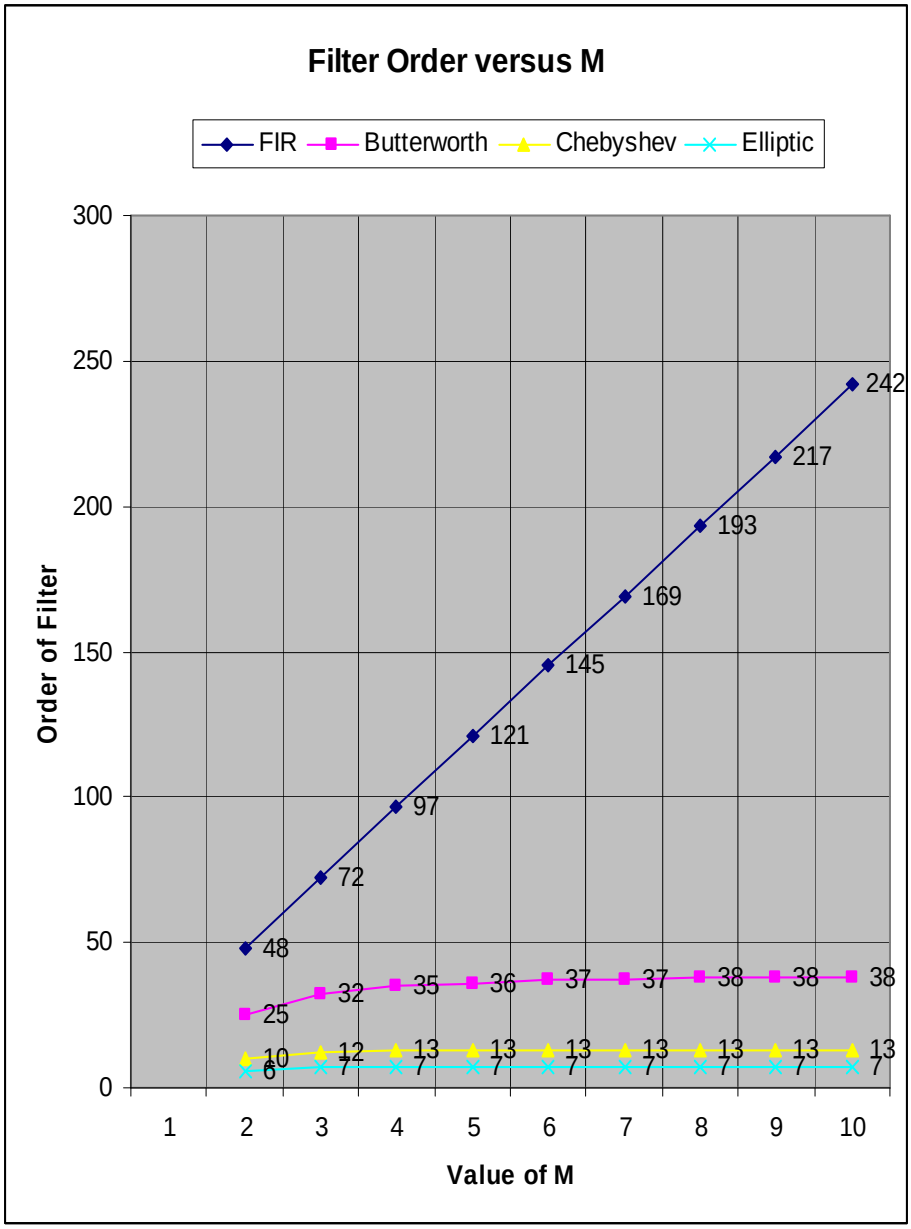


Fig.3. 15 Filter order required for different values of M.

From Fig.3.15, we find that Elliptical design of IIR filter provides the lowest order filter. The IIR filter has numerator as well as denominator in its transfer function so it requires 2N multiplications, where N is the order of the filter. Since elliptical design has the lowest order, so it is selected for comparison of computational costs [12].

If N is the order of FIR filter, then MN multiplications are required to get one

output sample, in the direct implementation. In case of polyphase implementation of an FIR filter, the filter is decomposed into M sub-filters each of length N/M. Here we require N multiplications to produce one output sample.

In case of conventional IIR filter of order N, we require (2N + 1)M multiplications to get one output sample. In the transformed IIR filter presented here, number of multiplications per output sample is (N + NM/2). The computational costs for each of the above filter architectures is calculated and shown in Table 3.3.

Table 3.3 Computational Costs of Various Architectures

	Order of	Comp. Cost of	Order of	Computational Cost of			% age Reduction compared with Polyphase	
M	**FIR (K)**	**Polyphase FIR**	**Elliptic IIR (N)**	**Conv. IIR (2N+1) M**	**Polyphase IIR (NM+M+1)**	**Trans. IIR (N+MN/ 2)**	**FIR**	**IIR**
2	48	48	6	26	15	12	75	20
3	72	72	7	45	25	17.5	75.7	30
4	97	97	7	60	33	21	78.35	36.36
5	121	121	7	75	41	24.5	79.75	40.24
6	145	145	7	90	49	28	80.68	42.85
7	169	169	7	105	57	31.5	81.36	44.74
8	193	193	7	120	65	35	81.86	46.15
9	217	217	7	135	73	38.5	82.25	47.26
10	242	242	7	150	81	42	82.64	48.15

The computational costs are plotted in Fig.3.16. From the figure, it is clear that transformed IIR filter has the lowest computational cost. The reduction in cost is 82.64% as compared to FIR decimation filters for M = 10. As compared to polyphase IIR structures, the reduction of the order of 48% is achieved. The computational cost reduces further for higher values of M. This shows that an

efficient decimator structure is obtained.

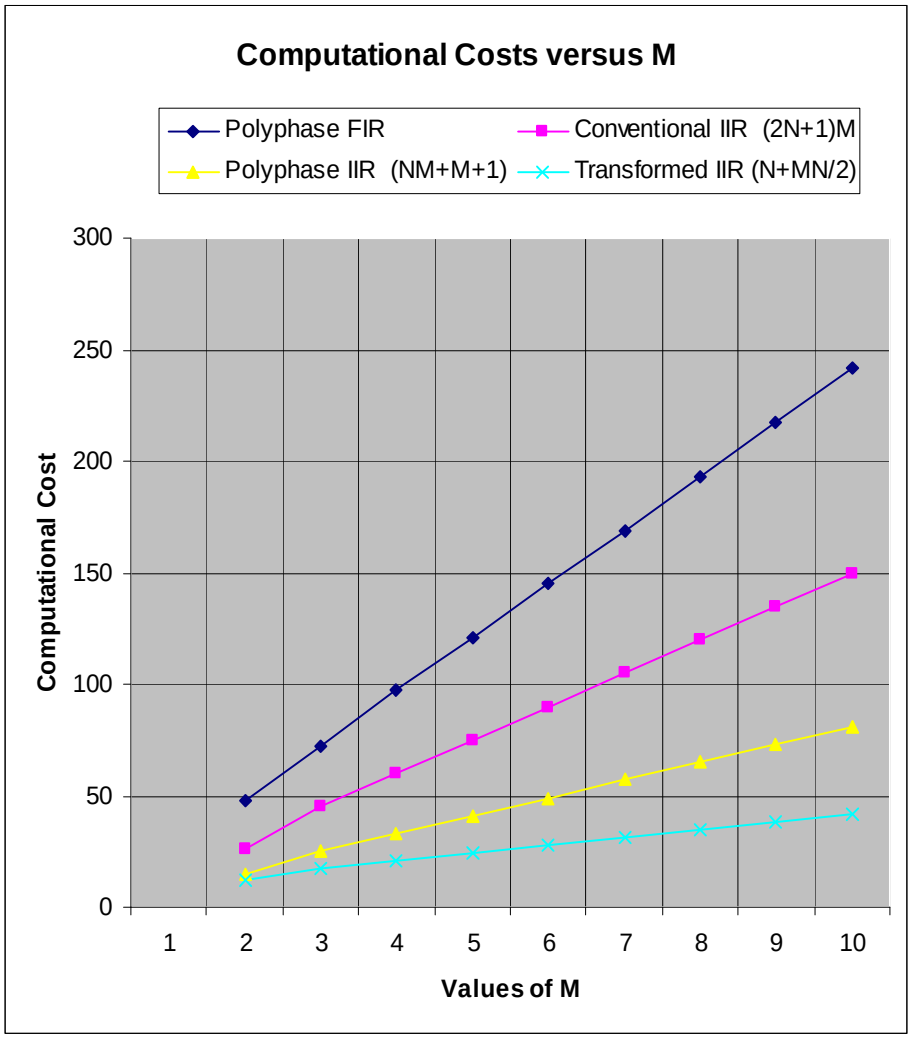


Fig.3. 16 Computational Cost for various filter architectures

3.7 Conclusions

Transformation of first order IIR filter into an efficient decimation filter was described. Efficient architectures for first order recursive filters and second order recursive filters were derived and implemented. Frequency responses of the transformed filters were studied. It was observed that frequency response plots of the transformed filters have good agreement with original filters. Decimation and filtering was achieved in the same stage. Computational costs of the transformed filters were compared with conventional IIR/FIR decimation filters.

Reduction in computational cost was observed that increases with higher values of M.

3.8 *References*

[1] Hassan Aboushady, Y. Dumonteix, M.M. Louerat and H. Mehrez, “Efficient Polyphase Decomposition of Comb Decimation Filters in sigma-delta Analog-to-Digital Converters,” IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing, Vol.48, No.10, October 2001, Page(s): 898-903.

[2] Tecpanecatl-Xihuitl, J.L., Kumar, A., and Bayoumi, M.A.**, “**Low complexity decimation filter for multi-standard digital receivers,” IEEE International Symposium on Circuits and Systems, ISCAS 2005, Vol.1, 23-26 May 2005, Page(s):552 – 555.

[3] Alan V. Oppenheim and Ronald W. Schafer, “Discrete-Time Signal Processing“, Prentice Hall of India, ch.6, 2001

[4] Vivek Venugopal, Khalid H. Abid, and Shailesh B. Nerurkar,”Design and Implementation of a Decimation Filter for Hearing Aid Applications,” Proc. IEEE Southeast Conf. 2005, April 2005, Page(s):111-115.

[5] P.P. Vaidyanathan, “Multirate Systems and Filter Banks”, Pearson Education, Inc, ch.4, 2004.

[6] fred j. harris and Benjamin Egg, ”Forming Narrowband Filters at a fixed Sample Rate with Polyphase Down and Up Sampling Filters,” Proc. of the 2007 Intl. Conf. on Digital Signal Processing (DSP 2007), 2007, Page(s):315-318,.

[7] Miljana Sokolovic, Borisav Jovanovic, and Milunka Damnjanovic, “Decimation Filter Design,” Proc. 24th International Conference on Micoelectronics (MIEL 2004), Vol.2, Nis, Serbia and Montenegro, 16-19 May 2004, Page(s):601-604.

[8] frederic j. harris, C. Dick, and M. Rice, “Digital Receivers and Transmitters using Polyphase Filter Banks for Wireless Communications,” IEEE Transactions on Microwave Theory and Techniques, vol.51, No.4, April 2003, Page(s):1395-1412.

[9] A. Krukowski, I. Kale and G.D. Cain, “Decomposition of IIR Transfer Function into Parallel Arbitrary-Order IIR Subfilters”, IEEE Nordic Signal Processing Symposium (NORSIG’96), Sep 1996, Page(s):1-9.

[10] Martinez, H., and Parks, T., ”A class of infinite-duration impulse response digital filters for sampling rate reduction,” IEEE Transactions on Acoustics, Speech, and Signal Processing , Volume 27, Issue 2, Apr 1979, Page(s):154 – 162.

[11] Hakan Johansson and Lars Wanhammar, “High-Speed Recursive Filter Structures Composed of Identical All-Pass Subfilters for Interpolation, Decimation, and QMF Banks with Perfect Magnitude Reconstruction,” Trans. IEEE Trans. on Circuits and Systems-II: Analog and Digital Signal Processing, Vol.46, No.1, January 1999, Page(s):16-28.

[12] Charles M. Rader, “The Rise and Fall of Recursive Digital Filters, “IEEE Signal Processing Magazine, Nov. 2006, Page(s):46-49.

……..

CHAPTER- 4

EFFICIENT ARCHITECTURES FOR INTERPOLATION FILTERS

4.1 Introduction

In interpolation or up sampling, output sampling rate is higher than input. Zero-valued samples are inserted between two adjacent input samples [1, 2, 3]. The role of an interpolation filter is to preserve the input signal in the range [0, F_s] where F_s is sampling frequency and to eliminate the extra images as good as desirable. From noble identity of multirate systems, interchange of upsampler and filter is possible [4 – 7]. The merged delay transformation is used to convert recursive filter transfer function into an efficient interpolation filter. Efficient architectures for IIR interpolation filters are presented in this chapter.

Transformation of first order IIR filter into an efficient interpolation filter is described in section 4.2. Relevant equations are derived and efficient architectures are presented. Section 4.3 describes the transformation of second order IIR filter into an efficient interpolation filter. In general, transformation of any order IIR filter into an interpolation filter is described in section 4.4. Frequency response of transformed filters is compared with the original filters in section 4.5. Computational costs are compared in section 4.6.

4.2 Transformation of First Order IIR Filter into an Efficient Interpolation Filter

In case of interpolation, the operation shown in Fig.4.1 is desired [5].

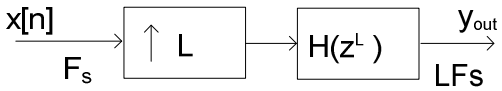


Fig.4. 1 Block diagram of interpolation filter

Here F_s is the sampling frequency at the input, L is an integer and LF_s is the output sampling frequency. Since by merged delay transformation we are able to convert H(z) into $H^{/}(z^L)$ so noble identity of multirate signal processing can be invoked and the up sampler may be shifted after the filter. In this way, an equivalent structure of Fig.4.2 is obtained [8 – 12].

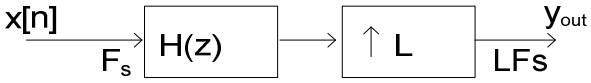


Fig.4. 2 Up sampler shifted after the filter using noble identity.

As clear from Fig.4.2, the filter H(z) instead of $H(z^L)$ precedes the 1-to-L up sampler.

Equation of Merged Delay Transformation derived in Chapter 2, is used with M substituted by L and we obtain as follows.

$$y[n] = p_1^L y[n - L] + \sum_{k=0}^{L-1} p_1^k r_1 \, x[n - k] \qquad \text{Eq(4.1)}$$

Here L is an integer factor. This is up-sampling factor for the interpolation filter. Taking Z-Transform of both sides of Eq(4.1), the transfer function H(z) is obtained as follows.

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{L-1} p_1^k r_1 z^{-k}}{1 - p_1^L z^{-L}} \qquad \text{Eq(4.2)}$$

Here the terms in the denominator represent the recursive part and the terms in the numerator are like an FIR filter.

H (z) can be expressed in the following manner.

$$H(z) = \frac{1}{1 - p_1^L z^{-L}} \left[r_1 + r_1 p_1 z^{-1} + r_1 p_1^2 z^{-2} + \ldots + r_1 p_1^{L-1} z^{-(L-1)} \right] \qquad \text{Eq(4.3)}$$

Equation (4.3) can be implemented as shown in Figure 4.3.

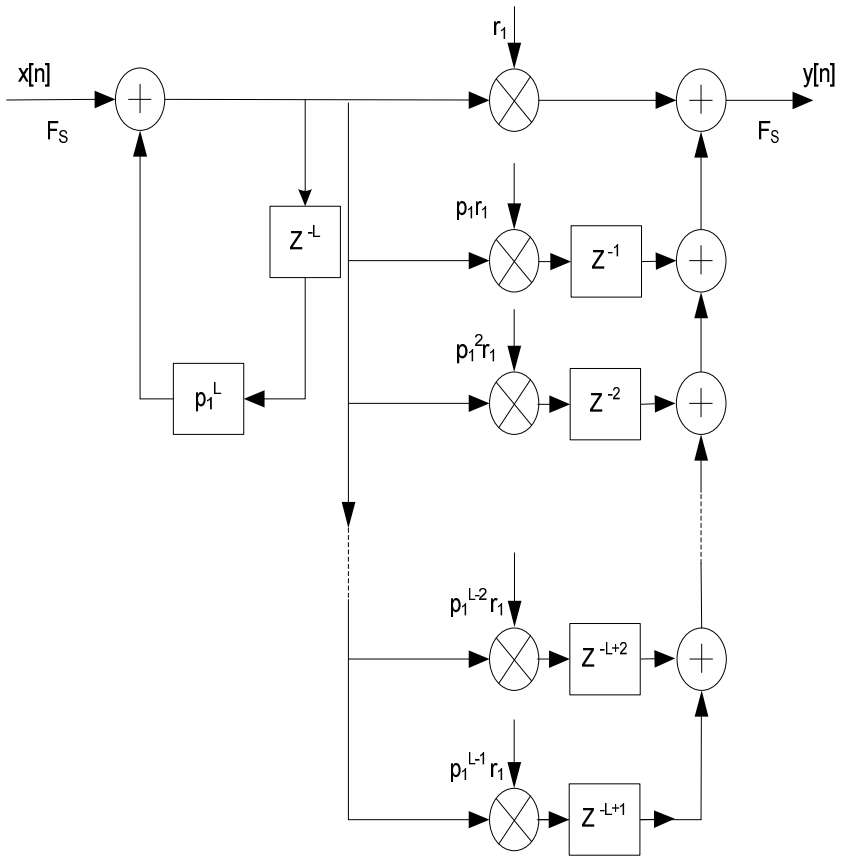


Fig.4. 3 Direct implementation of Eq(4.3).

Sampling rate is same at the input and output of Fig.4.3. However this structure is suitable for sample rate conversion. It can be easily transformed into an efficient interpolation filter by the following process.

Figure 4.3 is L-path parallel decomposition that is like poly phase structure. If up-sampling is desired, we have to introduce 1-to-L up-sampler at the input of Fig.4.3, as shown in Fig.4.4.

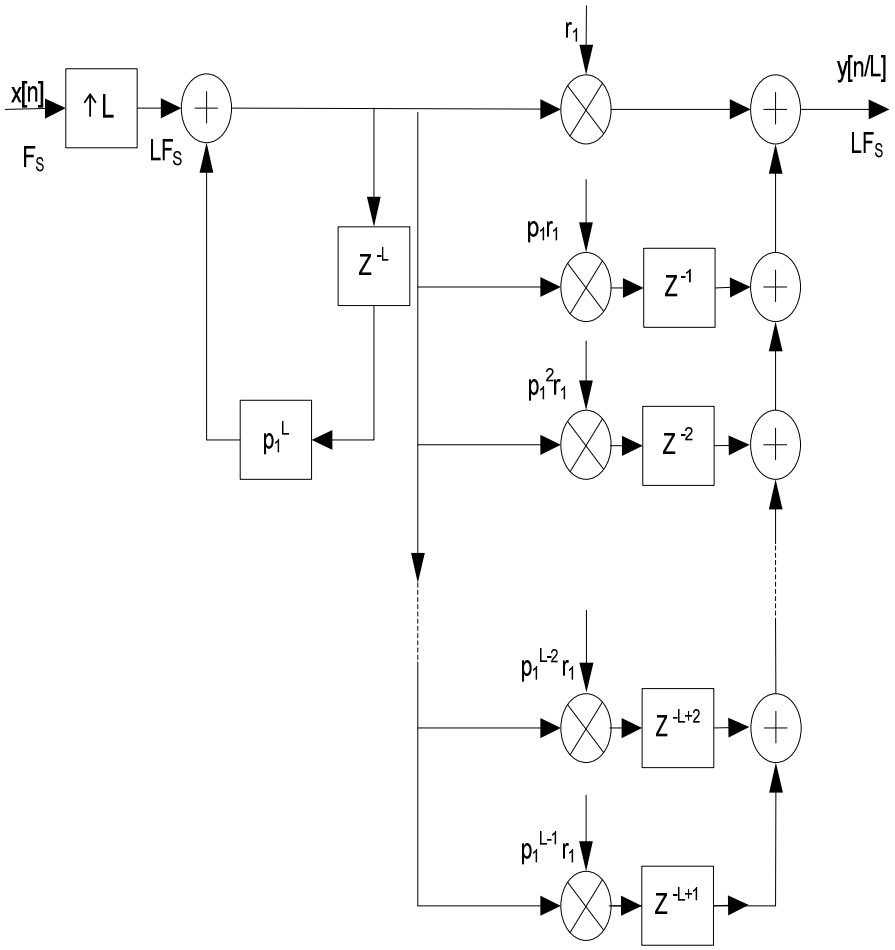


Fig.4.4 Up-sampler introduced at the input

When noble identity is applied to the IIR part of the transfer function, the 1-to-L up-sampler is shifted after the recursive part by changing Z^{-L} to Z^{-1} as shown in Fig.4.5.

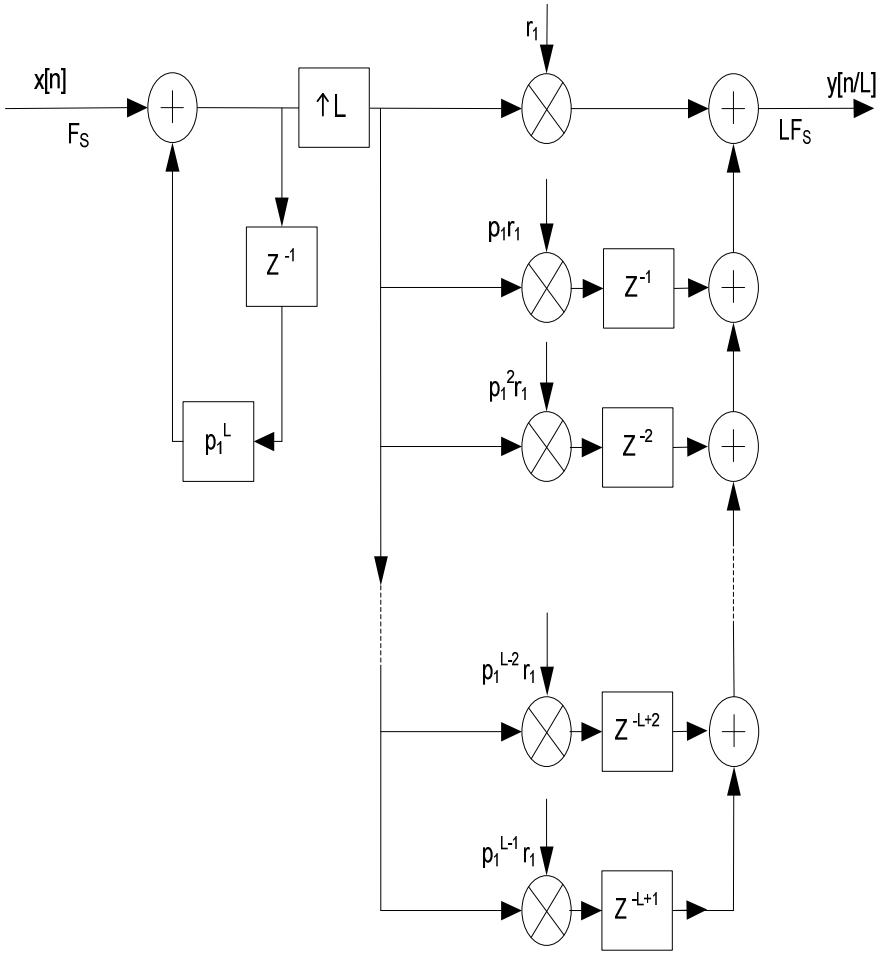


Fig.4. 5 Shifting the up-sampler by applying noble identity.

The up-sampler is further shifted in each of the L-paths as shown in Fig.4.6.

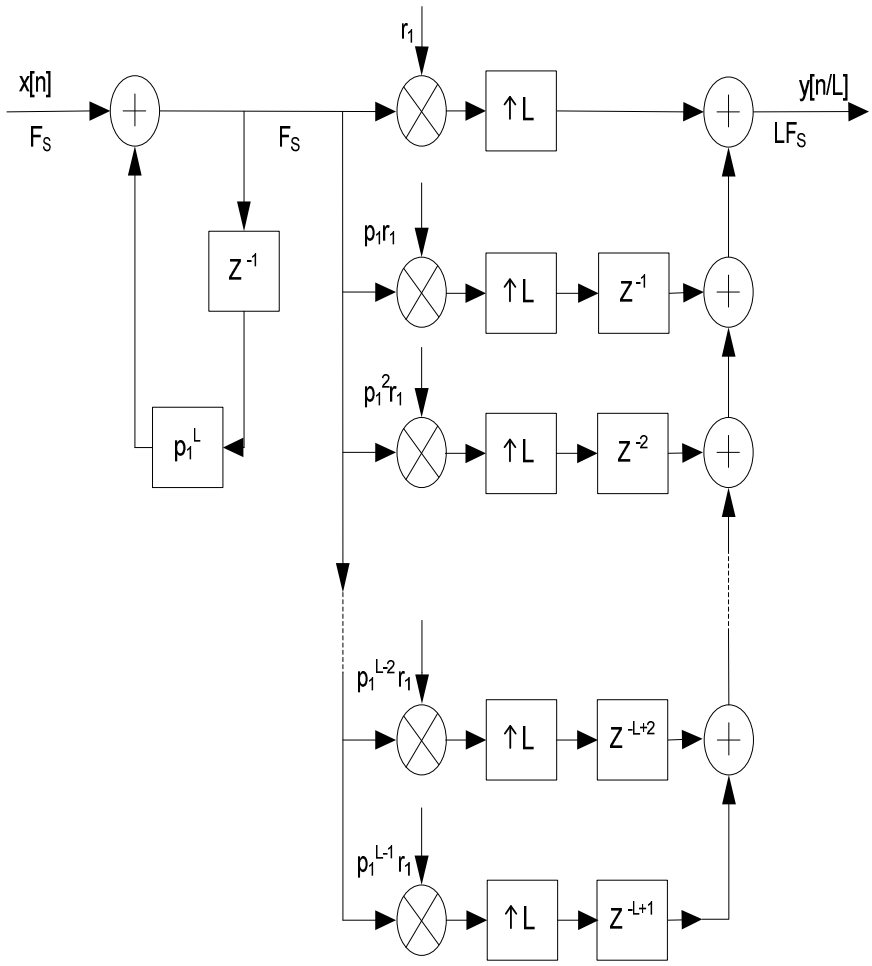


Fig.4. 6 Shifting of 1-to-L up-sampler into L-parallel paths.

Up-samplers associated with increasing delays are implemented by a commutator switch model moving in clockwise direction. The efficient architecture for transformed first order IIR filter is obtained, as shown in Fig.4.7.

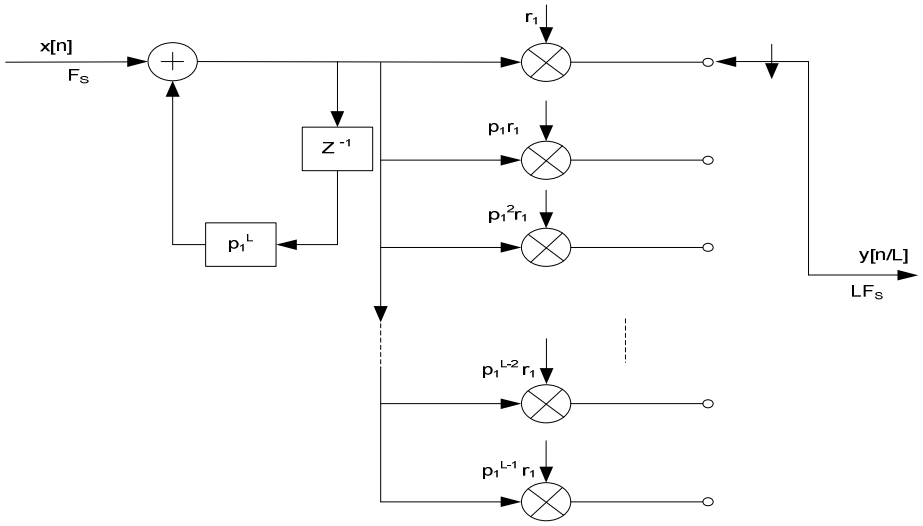


Fig.4. 7 Efficient Architecture for first order IIR Filter transformed into 1-to-L Interpolation Filter

In this way, (L – 1) samples are inserted between two adjacent samples and 1-to-L interpolation is achieved.

4.3 Transformation of Second Order IIR Filter

A second order system is decomposed into the first order parallel sections and following the procedure of section 2.2.2, the transfer function $H_{1R}(z) = Y_{1R}(z)/X(z)$ is obtained as follows.

$$H_{1R}(z) = \frac{(1 - A_{LR} z^{-L}) \sum_{k=0}^{L-1} z^{-k} B_{kR} - A_{LI} z^{-L} \sum_{k=0}^{L-1} z^{-k} B_{kI}}{1 - 2A_{LR} z^{-L} + (A_{LR}^2 + A_{LI}^2) z^{-2L}} \qquad \text{Eq(4.4)}$$

Here the values of A_{LR}, A_{LI}, B_{kR} and B_{kI} are given below.

$$A_{LR} = \text{real}(p_1)^L \qquad A_{LI} = \text{imag}(p_1)^L$$
$$B_{kR} = \text{real}(r_1(p_1)^k) \qquad B_{kI} = \text{imag}(r_1(p_1)^k) \qquad \text{Eq(4.5)}$$

The denominator part of $H_{1R}(z)$ is implemented first for convenience of application of noble identity of multirate systems. The numerator of $H_{1R}(z)$ can be implemented in L-parallel sub-filters with increasing delays. Each sub-filter should be having two non-zero coefficients; [B_{kR} , (L-1) zeros, ($-A_{LR}B_{kR} - A_{LI}B_{kI}$)] where $0 \le k \le L-1$. We can express $H_{1R}(z)$ in the following form.

$$H_{1R}(z) = \frac{\sum_{k=0}^{L-1} H_k(z^L) z^{-k}}{1 - A_F z^{-L} - B_F z^{-2L}} \qquad \text{Eq(4.6)}$$

Here

$$A_F = 2A_{LR} \qquad B_F = -(A_{LR}^2 + A_{LI}^2)$$

The transfer function $H_{1R}(z)$ is implemented as shown in Fig.4.8.

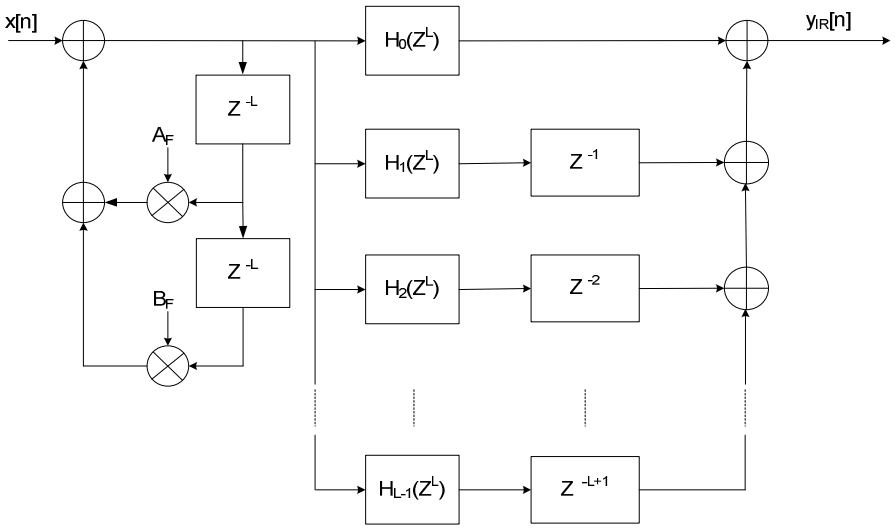


Fig.4.8 Simple Implementation of Eq(4.6)

To convert above structure into an interpolation filter, a 1-to-L up-sampler is introduced at the input as shown in Fig.4.9.

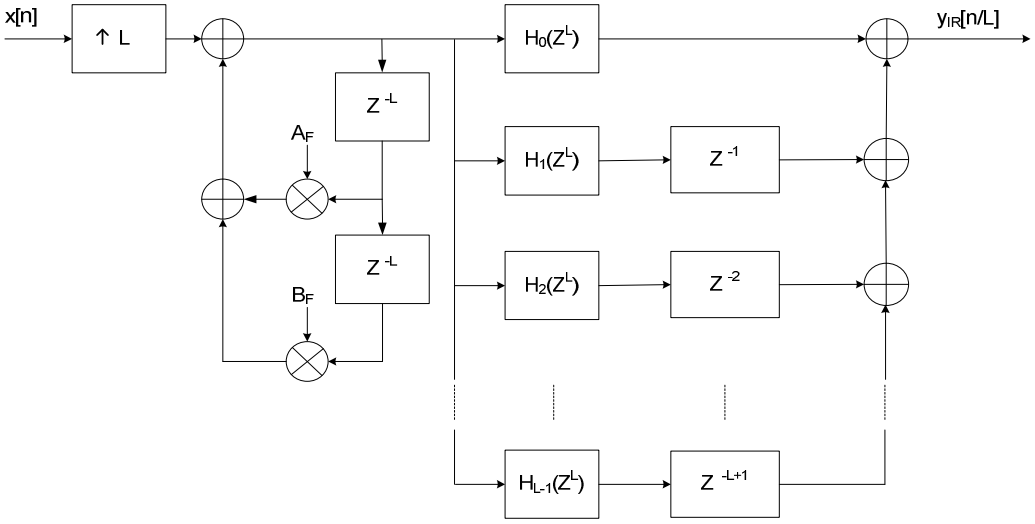


Fig.4.9 Up sampler attached at the input side.

The noble identity is applied to the structure of Fig.4.9 and up-sampler is shifted to the right of recursive part as shown in Fig.4.10.

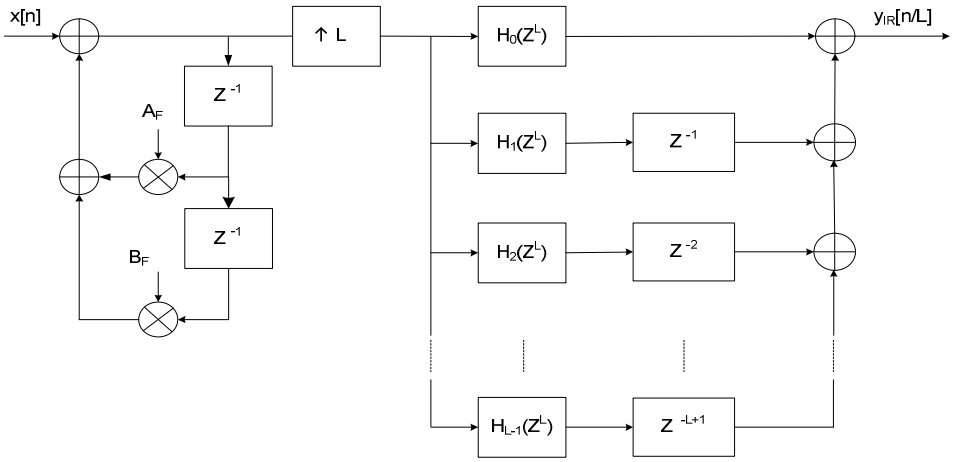


Fig.4. 10 Shifting right the up-sampler using noble identity

Again, the up-sampler is shifted to each branch as shown in Fig.4.11.

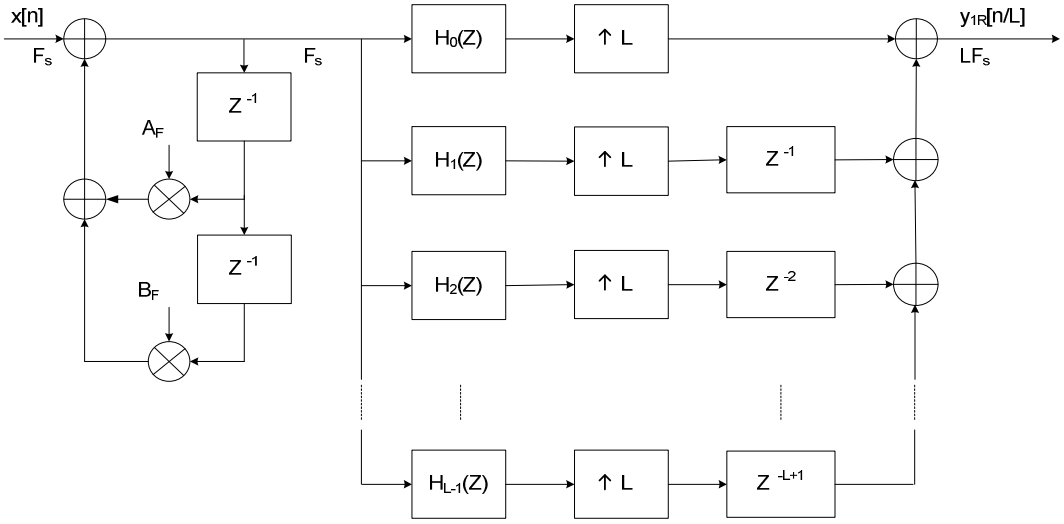


Fig.4. 11 Up-sampler shifted to each parallel path.

Up-samplers with increasing delays are implemented by commutator switch model rotating in clockwise direction as shown in Fig.4.12.

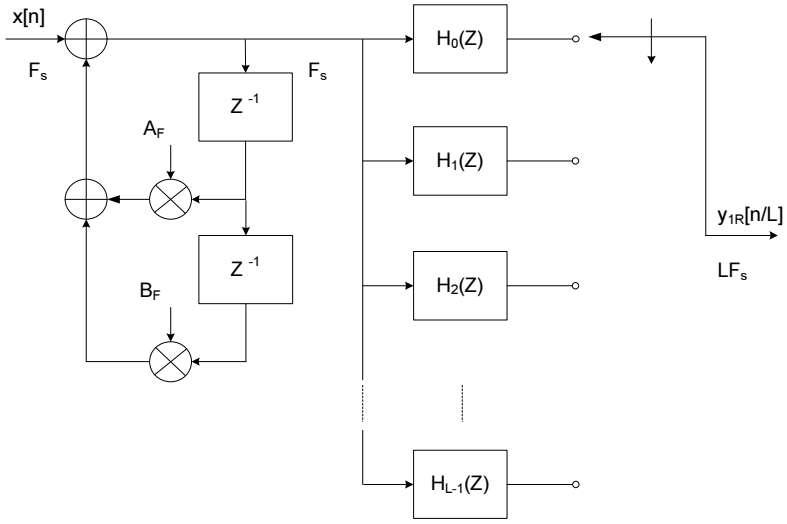
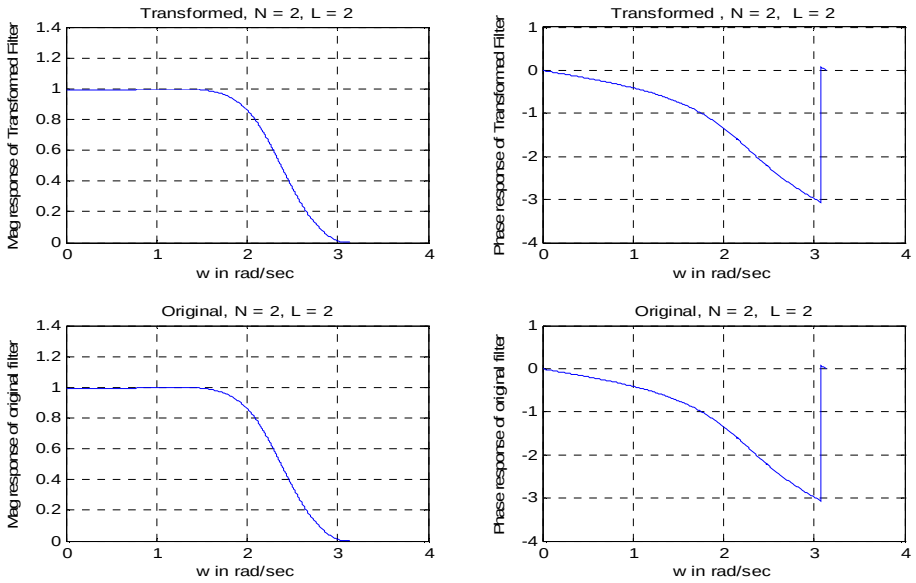


Fig.4. 12 Efficient Architecture of second order IIR filter transformed into interpolation filter

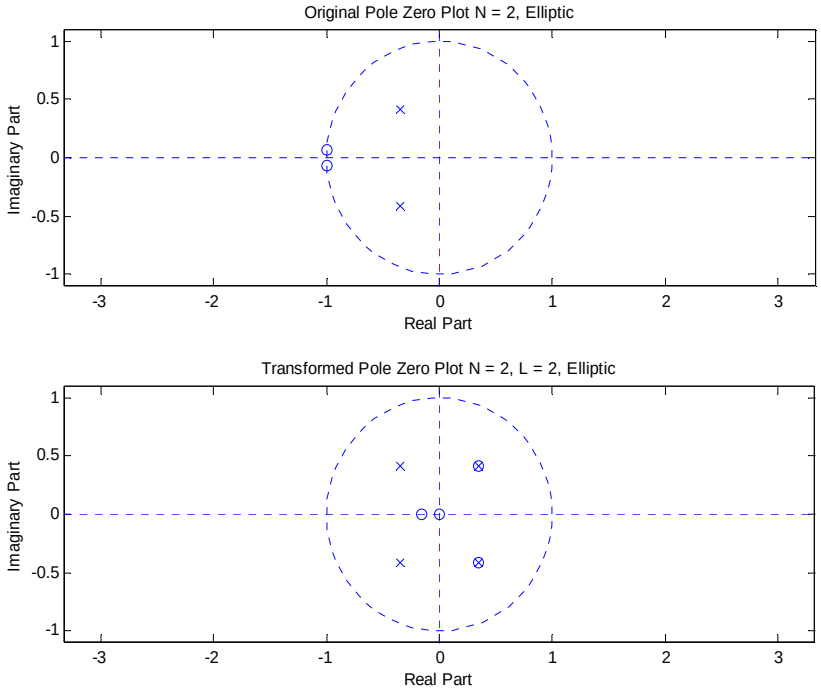
Implementation of Eq(4.6) resulted in an efficient architecture as imaginary outputs were not required in the final result. Two real outputs are equal so computation of one real output is sufficient.

4.4 Frequency Response and Pole-Zero Plots of Transformed Filters

The transfer function, representing real part of a second order section $H_{1R}(z)$ was checked for pole-zero positions and frequency response. It was compared with the original filter. The results for L = 2, L = 4 and L = 8 using N = 2 are shown in Fig.4.13, 4.14 and 4.15.

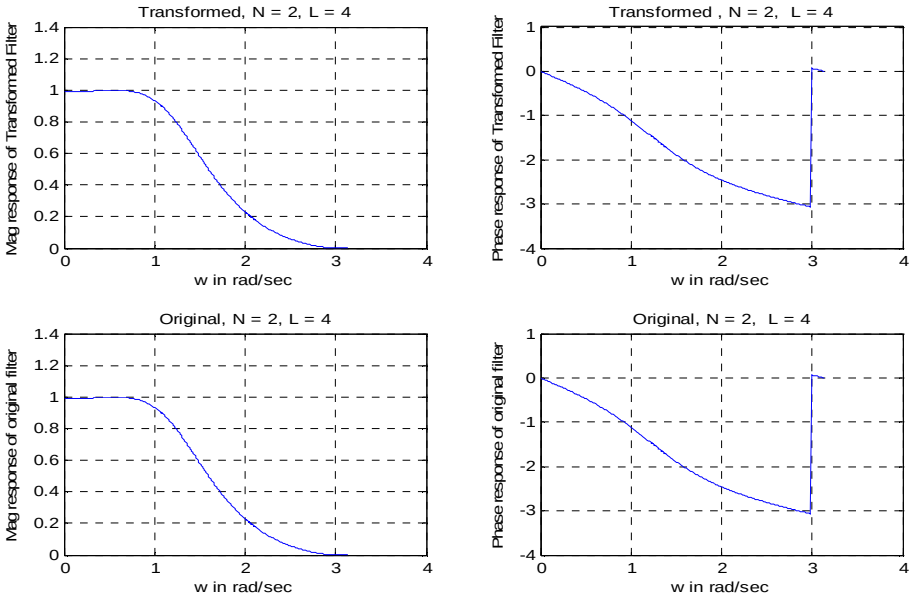


Frequency Response

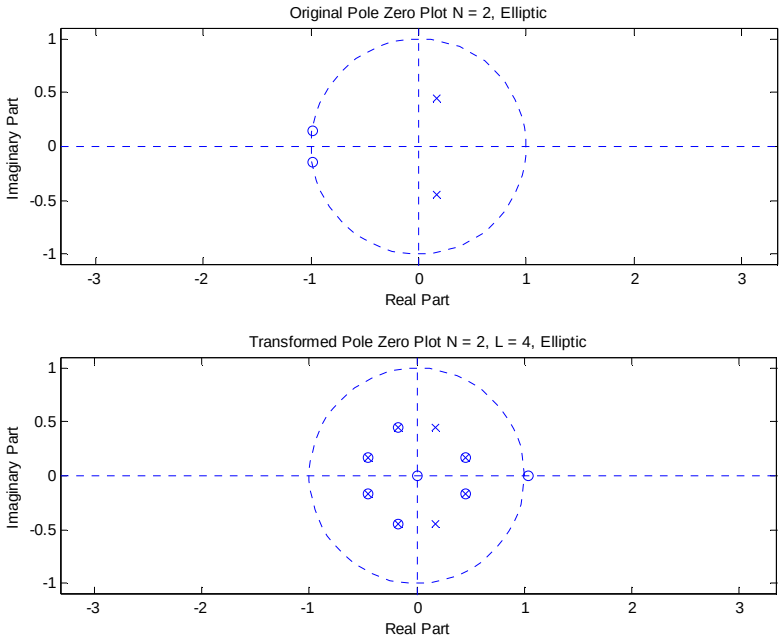


Pole-Zero Plots

Fig.4. 13 Frequency response and pole-zero plots for N = 2, L = 2.

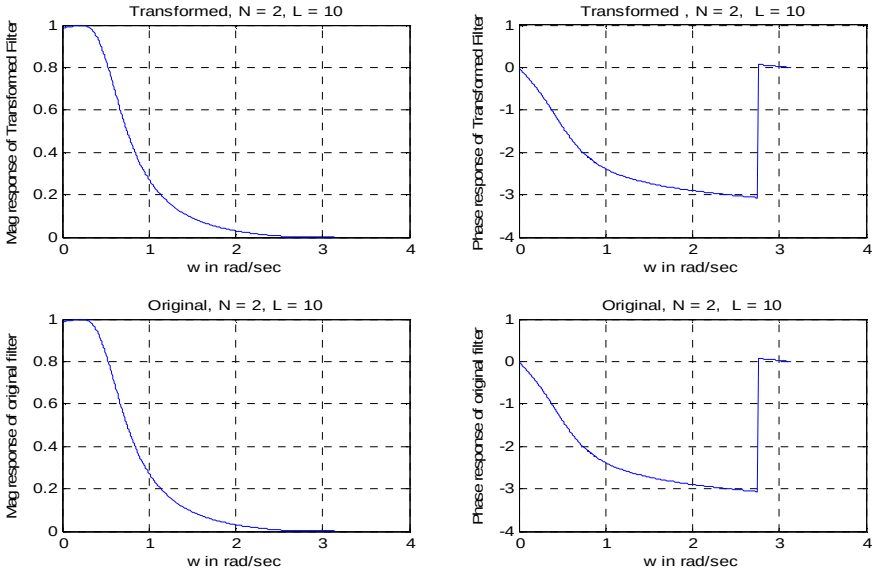


Frequency response

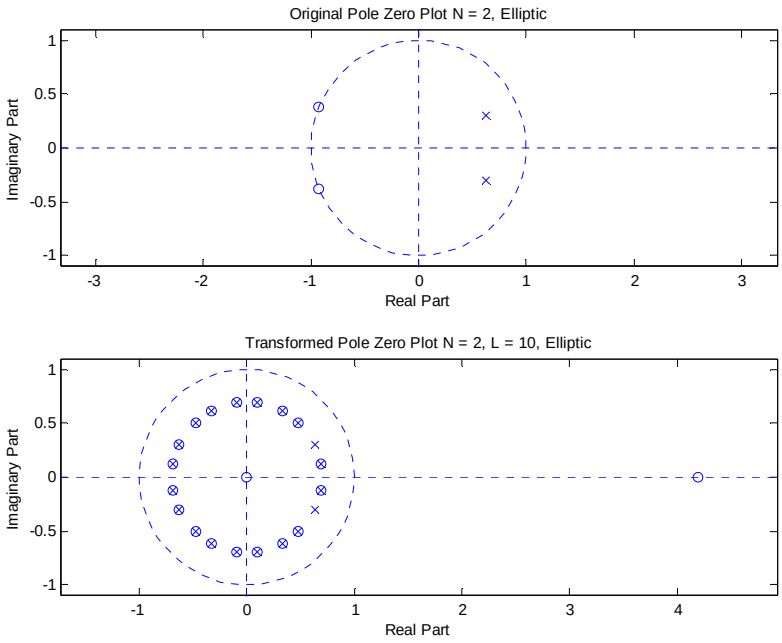


Pole-Zero Plots

Fig.4. 14 Frequency response and pole-zero plots for N = 2, L = 4.



Frequency response



Pole-Zero Plots

Fig.4. 15 Frequency response and pole-zero plots for N = 2, L = 10.

From pole-zero plots we see that additional pole-zero pairs are introduced due to merged delay transformation. The number of such pole-zero pairs depends on L. Total number of poles including the original poles (two in this case) becomes 2L. Frequency response curves show that the frequency response of the transformed filter is same as the original filter.

4.5 Computational Costs

In this section we compare the computational costs of various implementations of IIR interpolation filters, measured in multiplications per output sample. We use the same filter specifications as given Table 3.1.

In case of interpolation, where low pass filter is followed by an 1-to-L up sampler, the filter's frequency edges must satisfy the following relationships.

$$f_{passband} = \frac{0.45F_s}{L}$$

and

$$f_{stopband} = \frac{0.55F_s}{L} \qquad \text{Eq(4.7)}$$

We consider the implementation of this interpolation filter using conventional IIR filters, transformed IIR filters and polyphase IIR filters. Each of these filters are designed to meet the specifications of Table 3.1. The pass-band and stop-band frequencies were calculated for various values of L, using Eq.(4.7). The minimum required filter orders obtained by Matlab are shown in Table 4.1.

Table 4. 1 Filter Orders for various implementations from Matlab

Filter Type	**Value of L**				
	2	**4**	**6**	**8**	**10**
IIR Butterworth	25	35	37	38	38
IIR Chebyshev	10	13	13	13	13
IIR Elliptical	6	7	7	7	7
FIR	48	97	145	193	242

From Table 4.1, we find that Elliptical design of IIR filter provides the lowest order. Out of conventional IIR filters, we choose the computationally efficient elliptical IIR design for comparison.

If K is the order of FIR filter, then KL multiplications are required to get one output sample, in the direct implementation. In case of polyphase implementation of an FIR filter, the filter is decomposed into L sub-filters each of length K/L [12]. Here we require K/L multiplications to produce one output sample.

In case of conventional IIR filter of order N, we require 2N multiplications to get one output sample. In the transformed IIR filter presented here, number of multiplications per output sample is (N + N/L). The computational costs for each of the above filter architectures is shown in Table-4.2.

Table 4. 2 Computational costs of various architectures

	Order of	Cost of	Order of	Computational cost of			
L	FIR (K)	Polyphase FIR (K/L)	Elliptic IIR N	Conv. IIR (2N)	Trans. IIR (N+N/L)	% age Reduction as compared to IIR	% age Reduction as compared to FIR
2	48	24	6	12	9	25	62.5
3	72	24	7	14	9.33	33.36	61.12
4	97	24.2	7	14	8.75	37.5	63.92
5	121	24.2	7	14	8.4	40	65.29
6	145	24.1	7	14	8.17	41.64	66.1
7	169	24.1	7	14	8	42.86	66.8
8	193	24.1	7	14	7.87	43.78	67.34
9	217	24.1	7	14	7.77	44.5	67.33
10	242	24.2	7	14	7.7	45	68.18

In Table 4.2, the reduction in cost is as compared to conventional IIR implementation. The reduction for L = 10 is of the order of 45% which increases with L. The reduction in cost is about 68% as compared to poly phase FIR. This

data is plotted in Fig.4.16.

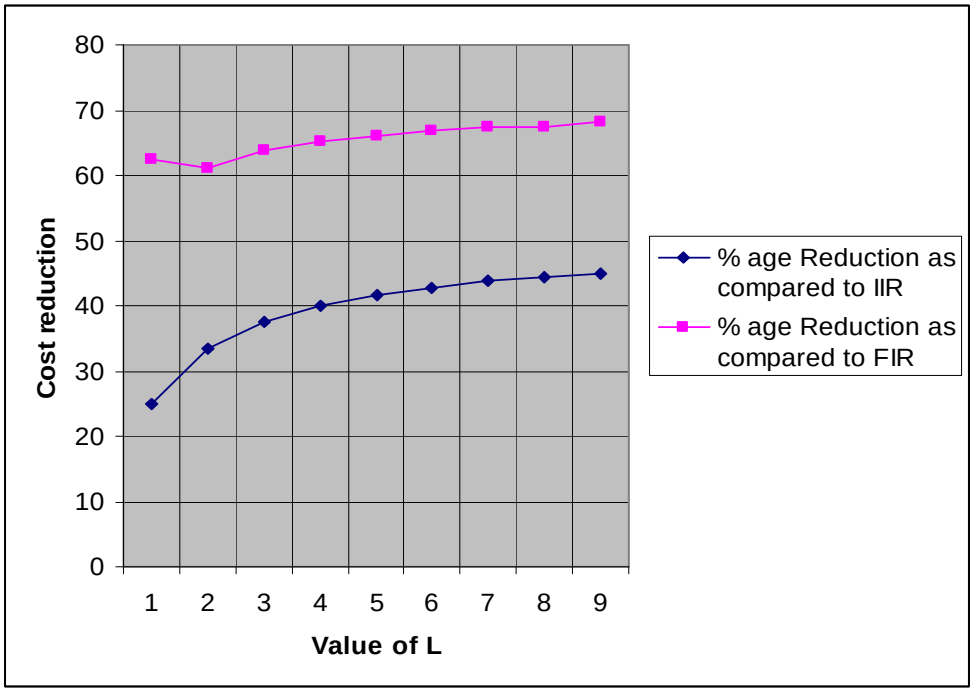


Fig.4.16 Reduction in computational cost

From Fig.4.16, we see that transformed filter provides an efficient architecture.

4.6 Conclusions

Transformation of first order IIR filter into an efficient interpolation filter was described. Efficient architectures for first order recursive filters and second order recursive filters were derived and implemented. Computational costs of the transformed filters were calculated. Reduction in computational cost up to 68% was achieved for L = 10. More reduction is possible with higher values of L.

Frequency responses of the transformed filters were studied. It was observed that frequency response plots of the transformed filters have good agreement with original filters. Interpolation and filtering was achieved in the same stage. This transformation can be applied to any order recursive filter.

4.7 References

[1] Mark W. Coffey, “Optimizing Multistage Decimation and Interpolation Processing- Part II”, IEEE Signal Processing Letters, Vol.14, No.1, January 2007, pp. 24-26.

[2] Charles M. Rader, “The Rise and Fall of Recursive Digital Filters,” IEEE Signal Processing Magazine, Nov. 2006, pp.46-49.

[3] Alan V. Oppenheim and Ronald W. Schafer, “Discrete-Time Signal Processing“, Prentice Hall of India, ch.6, 2001.

[4] Hakan Johansson and Lars Wanhammar, “High-Speed Recursive Filter Structures Composed of Identical All-Pass Subfilters for Interpolation, Decimation, and QMF Banks with Perfect Magnitude Reconstruction,” Trans. IEEE Trans. on Circuits and Systems-II: Analog and Digital Signal Processing, Vol.46, No.1, pp.16-28, January 1999.

[5] P.P. Vaidyanathan, “Multirate Systems and Filter Banks”, Pearson Education, Inc., ch.4, 2004.

[6] Ronald E. Crochiere and Lawrence R. Rabiner, “Interpolation and Decimation of Digital Signals- A Tutorial Review”, Proc. of the IEEE, vol.69, No.3, March 1981, pp.300-331.

[7] Joyce Van de Vegte, “Fundamentals of Digital Signal Processing”, Ch.14, 2002, Pearson Education Ltd.

[8] Valenzuela R. A. and A. G. Constantinides, “Digital Signal Processing Schemes for efficient interpolation and decimation”, IEE Proceedings, pt. G, vol.130, no.6, pp.225-235, December 1983.

[9] Eugene B. Hogenauer, “An Economical Class of Digital Filters for Decimation and Interpolation”, IEEE Trans. on Acoustics, Speech and Signal Processing, vol.ASSP-29, No.2, April 1981, pp.155-162.

[10] fred j. harris and Benjamin Egg,”Forming Narrowband Filters at a fixed Sample Rate with Polyphase Down and Up Sampling Filters,” Proc. of the 2007 Intl. Conf. on Digital Signal Processing (DSP 2007), pp.315-318, 2007.

[11] David J. Goodman and Michael J. Carey, “Nine Digital Filters for Decimation and Interpolation”, IEEE Trans. on Acoustics, Speech and Signal Processing, vol.ASSP-25, No.2, April 1977, pp.121-126.

[12] Bellanger, M., Bonnerot, G., and Coudreuse, M**.,** “Digital filtering by Polyphase network. Application to sample-rate alteration and filter banks;” IEEE Transactions on Acoustics, Speech, and Signal Processing, Volume 24, Issue 2, Apr 1976, pp.109 – 114.

…….

CHAPTER- 5

HARDWARE IMPLEMENTATIONS AND STABILITY ANALYSIS

5.1 Introduction

This chapter deals with the hardware implementation of the transformed filter. The filter architecture is described in Verilog HDL [1 – 5]. It is mapped on FPGA using Spartan-II technology [6 – 8]. Numerical analysis of finite word length used for implementation of filters is important [9, 10] in the hardware perspective. It provides information for number of bits required for satisfactory performance [11 – 13]. Similarly, coefficient quantization effect should be explored to ensure the stability of the filters [14 -16].

The Chapter is organized as follows. Section 5.2 describes the architecture and implementation results for a decimation filter. Section 5.3 describes the effect of coefficient quantization on pole-zero patterns by changing the number of bits used for implementation. Section 5.4 discusses the effect of transformation on the stability of the filters. Matlab simulations are used to investigate the issues.

5.2 Hardware Implementation

An audio interpolation filter was designed and implemented using the proposed architecture. To compare the implementation results, a polyphase FIR interpolation filter and cascaded IIR based interpolation filter was also implemented. The following specifications of the filter were used [11].

Input Sampling Frequency = 44.1 kHz

Passband Ripple < 0.1 dB

Stopband Attenuation = 96 dB

Interpolation Factor L = 4

Passband edge frequency = 20 kHz

Stopband edge frequency = 22.05 kHz

Using the Matlab FDA tool, FIR filter and IIR filter to meet the above specifications was designed. An FIR filter of 328 taps and IIR filter (Elliptic) of 13 order was required for the given specifications. The cascade IIR filter was implemented using direct form-II architecture. Hardware efficient architecture of polyphase FIR interpolator was used. All three architectures were first simulated in MATLAB, optimized in terms of coefficient lengths and then implemented on Virtex-5 FPGA using Verilog HDL. In order to estimate the power consumption of the architectures, Xilinx XPower tool was used. The implementation results of power and critical path delay for three architectures are plotted in Fig.5.1 and 5.2.

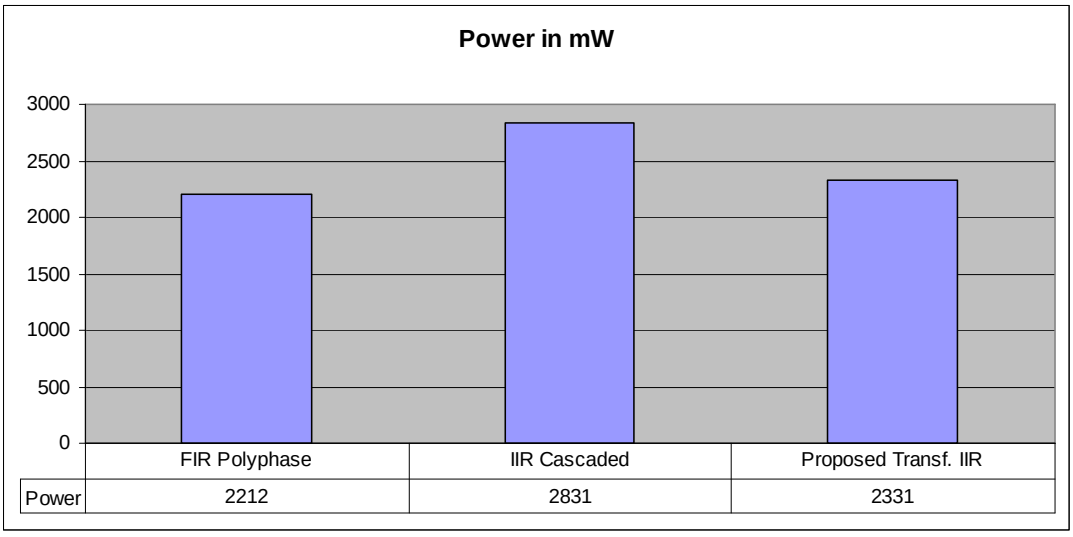


Fig. 5.1 Comparison of Power Consumption in mWatt

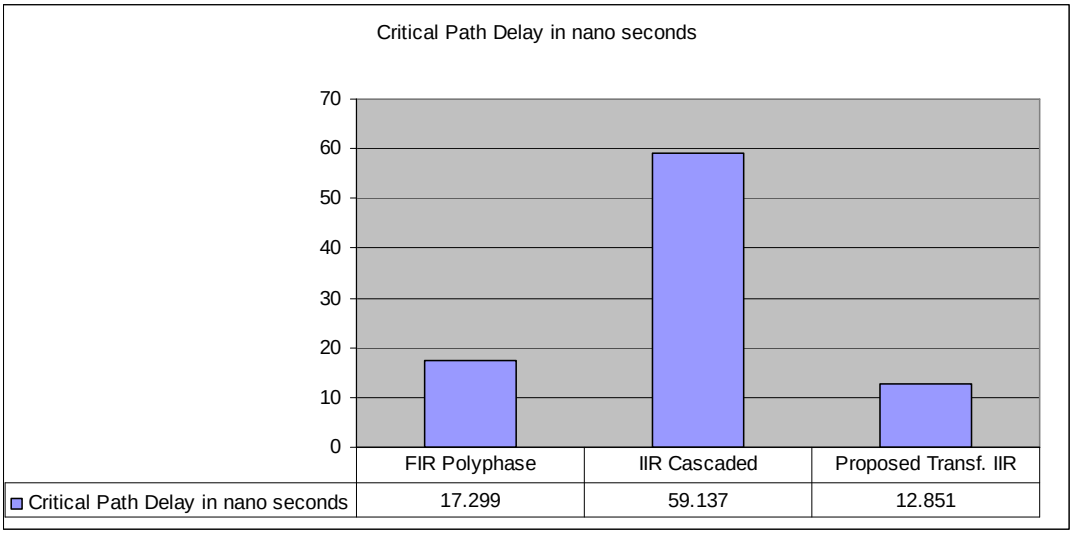


Fig.5. 2 Comparison of Critical Path Delay in nano seconds

As clear from the figure, the power consumption is lower than cascaded IIR based architecture. Its value is comparable to polyphase FIR architecture. The path delay is much smaller as compared to cascaded IIR architecture.

An IIR decimation filter was also implemented using this technique. The results for a first order IIR filter with a decimation factor of 4 are described. The filter is implemented using the architecture shown in Fig.5.3. The coefficients of the filter are obtained from Matlab and converted into Q1.15 format [In this format, MSB represents the sign bit and the next 15 bits represent the fractional part]. All components of the proposed decimation filter are specified in Verilog HDL. Simulation is carried out using Modelsim. The simulation results show that the hardware generates output after every (M-1) inputs.

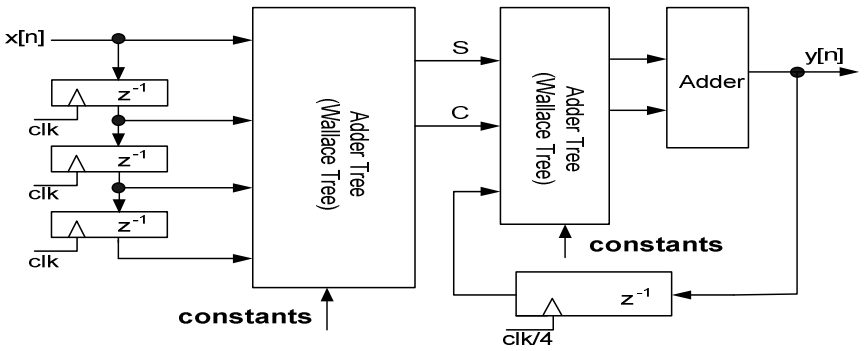


Fig.5.3 Hardware Architecture of first order IIR decimation filter

The hardware is synthesized on Xilinx FPGA using Spartan-II technology to determine the system size and speed. Multiplications of constants are carried out outside the hardware. Multiplications of constants with the variable data are implemented using adder tree (Wallace Tree). One adder is employed in the last stage to generate final output. The synthesis report is shown in Table-5.1. The hardware can operate at the clock frequency of 39. 403 MHz.

Table-5.1 Implementation Results

Selected Device : 2s200fg256-5

Number of Slices	**994 out of 2352 42%**
Number of Slice Flip Flops	364 out of 4704 7%
Number of 4 input LUTs	1908 out of 4704 40%
Number of bonded IOBs	50 out of 180 27%
Number of GCLKs	1 out of 4 25%
Maximum Frequency	39.403MHz

5.3 Effect of Merged Delay Transformation on Stability

Stability testing of an IIR filter is very important. Bounded Input Bounded Output (BIBO) stability condition requires that the impulse response is absolutely

summable i.e. $\sum_{n=-\infty}^{\infty} |h(n)| < \infty$. For a causal IIR filter, the impulse response extends to infinity so an alternate approach based on location of poles of the system can be used to check the stability. For an Nth order stable IIR transfer function, all the N poles must be strictly inside the unit circle in z-plane [17], [18].

In the proposed technique, a stable transfer function satisfying the filtering requirements is obtained from "Filter Design and Analysis" tool [19]. The higher order transfer function is decomposed into parallel first order and second order sections. The parallel structure is much less sensitive to coefficient quantization [17], [20]. For implementation as interpolator, the parallel sections are transformed using merged delay transformation. The transfer function for the first order section with real coefficients is transformed as;

$$H(z) = \frac{1}{1 - p_1^L z^{-L}} \left[r_1 + r_1 p_1 z^{-1} + \ldots + r_1 p_1^{L-1} z^{-(L-1)} \right]$$

The roots of polynomial $\left(1 - p_1^L z^{-L}\right)$ are the L- poles and all poles lie on a circle within a radius of $|p_1|$ and centered at zero. Thus stability condition for first order sections can be stated as $|p_1| < 1$. In the stable transfer function $|p_1|$ is always less than 1 so the transformed transfer function is always stable.

For experimental analysis, Pole-zero plots of transformed first order filters were investigated for various values of L. Typical results for L = 8 using a first order Butterworth filter are shown in Fig.5.4. All the poles of transformed transfer function are well within the unit circle.

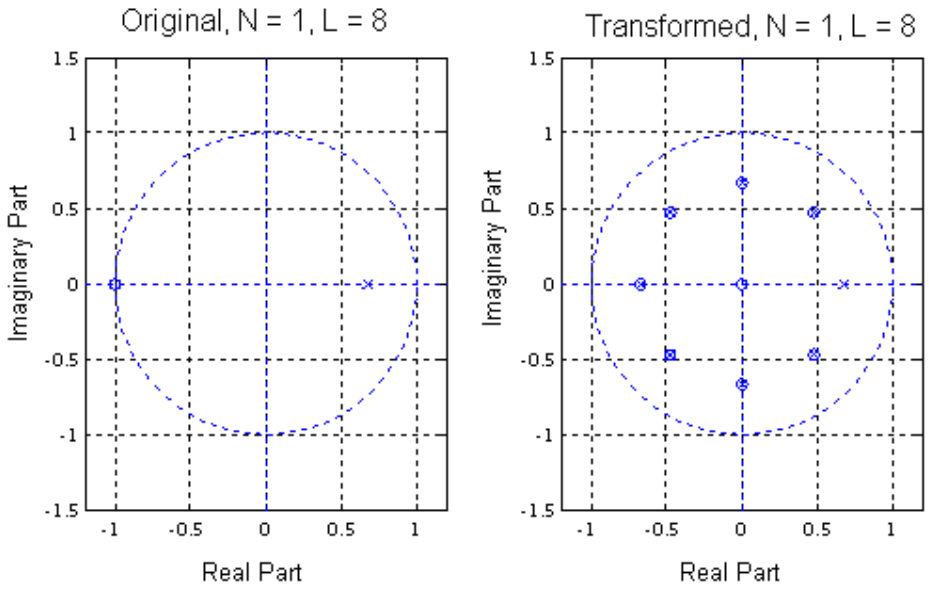


Fig.5.4 Pole-zero plots for original and transformed first order filter (N = 1, L = 8, Butterworth Filter)

The magnitude and phase responses for original and transformed filters are shown in Fig.5.5. The values of poles and zeros are shown in Table-5.2.

Table- 5. 2 Values of Poles and Zeros for N = 1, L = 8 Butterworth Filter

Original Filter		Transformed Filter	
Poles	Zeros	Poles	Zeros
0.6682	-1	-0.6682 -0.4725 + 0.4725i -0.4725 - 0.4725i -0.0000 + 0.6682i -0.0000 - 0.6682i 0.6682 0.4725 + 0.4725i 0.4725 - 0.4725i	0 0.4725 + 0.4725i 0.4725 - 0.4725i -0.6682 -0.4725 + 0.4725i -0.4725 - 0.4725i 0.0000 + 0.6682i 0.0000 - 0.6682i

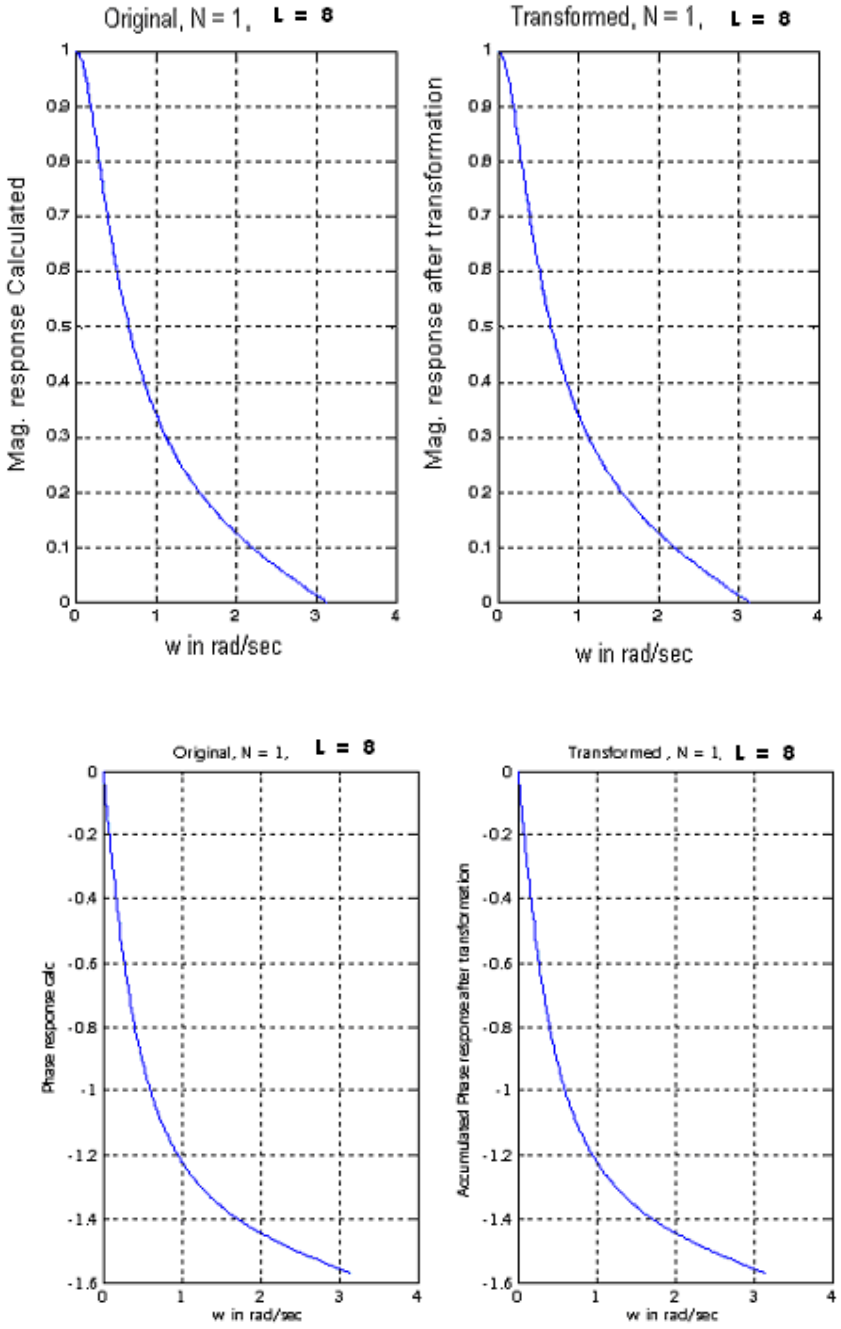


Fig.5.5 Magnitude and phase response of original and transformed filters

(N = 1, L = 8, Butterworth Filter)

The second order section with complex conjugate poles is transformed as:

$$H(z^L) = \frac{Numerator}{1 - 2A_{LR}z^{-L} + (A_{LR}^2 + A_{LI}^2)z^{-2L}}$$

Where $A_{LR} = real\left(p_1^L\right)$ and $A_{LI} = imag\left(p_1^L\right)$. Due to shifting of 1-to-L up sampler towards right side, the denominator of the transformed transfer function at the time of implementation becomes a second order polynomial in z i.e denominator = $1 - 2A_{LR}z^{-1} + (A_{LR}^2 + A_{LI}^2)z^{-2}$.

The coefficients of denominator of a second order transfer function should satisfy the following conditions to ensure stability [16].

$$\left|A_{LR}^2 + A_{LI}^2\right| < 1$$

and

$$\left|2A_{LR}\right| < 1 + \left(A_{LR}^2 + A_{LI}^2\right)$$

In our technique, a stable IIR filter is designed and then decomposed into parallel sections. For a real coefficient stable transfer function, real or complex poles are always within the unit circle. Now, if p_1 and p_2 are complex conjugate poles and p_1 = a +ib then $\left|p_1\right| = \left|p_2\right| = \left|\sqrt{a^2 + b^2}\right| < 1$. Higher powers of p_1 or p_2 are further away from the unit circle. In this way both of the above conditions are satisfied by the transformed transfer function.

Based on above conditions, the stability of transformed transfer functions for all types of IIR filters was studied based on pole-zero locations.

For the second order transfer function, the pole-zero plots are shown in Fig.5.6. It is observed that the proposed transformation introduces (L-1) number of additional poles within the unit circle. Equal number of zeros is also introduced that have the concurrent position with the additional poles.

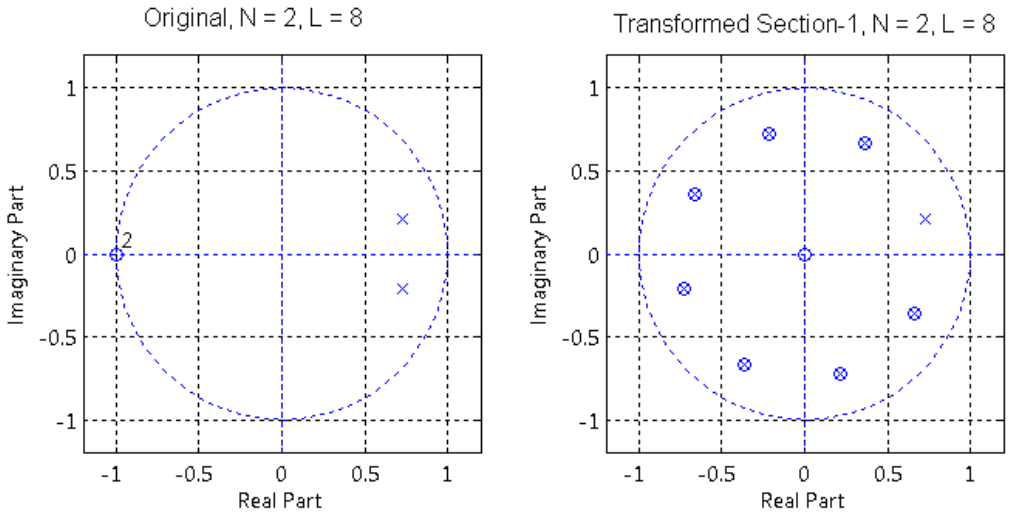


Fig.5.6 (a)

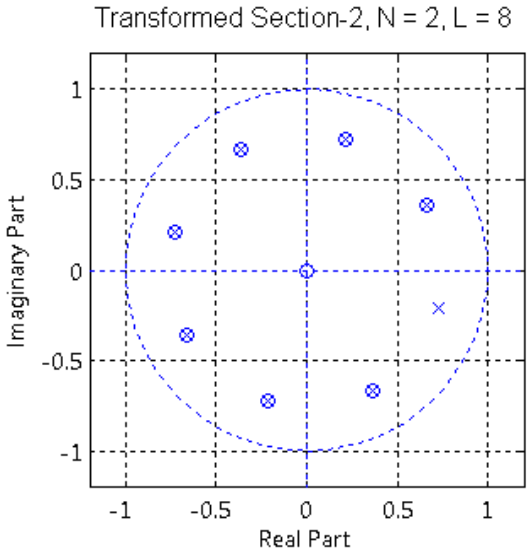


Fig.5.6(b)

Fig.5.6 (a) Pole-zero plots for original and transformed filter Section 1

(N = 2, L = 8, Butterworth Filter)

(b) Pole-zero plots for original and transformed filter Section 2

Stability of the system is not disturbed by increasing L. This is due to the fact that all pole-zero pairs introduced by transformation are inside the unit circle.

Moreover, the magnitude and phase responses are not changed by this transformation.

From Fig.5.4 and 5.6, it is observed that the proposed transformation introduces (L -1) number of poles within the unit circle. Equal number of zeros is also introduced at the additional pole locations so that the magnitude response is not disturbed. Here zeros do not lie on the unit circle. New poles and zeros are added exactly at the same positions inside the unit circle. This fact is also clear from the magnitude and phase response plots of the original and transformed filters as shown in Fig.5.7. The magnitude response of transformed filter is exactly same as the original filter for various values of L. Stability of the system is not disturbed since the pole-zero pairs introduced by increasing L are all within the unit circle. However, for increasing values of L, the location of pole-zero pair moves away from the centre of unit circle.

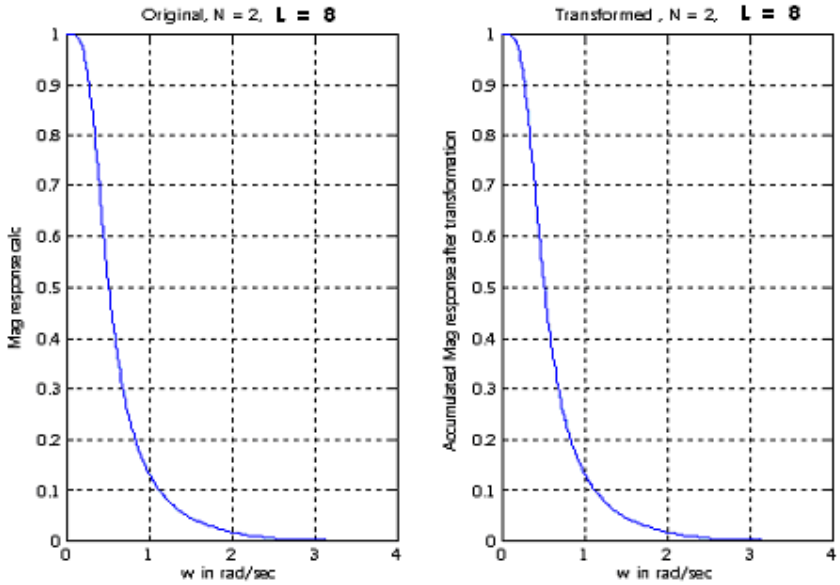


Fig.5.7 (a) Magnitude Responses

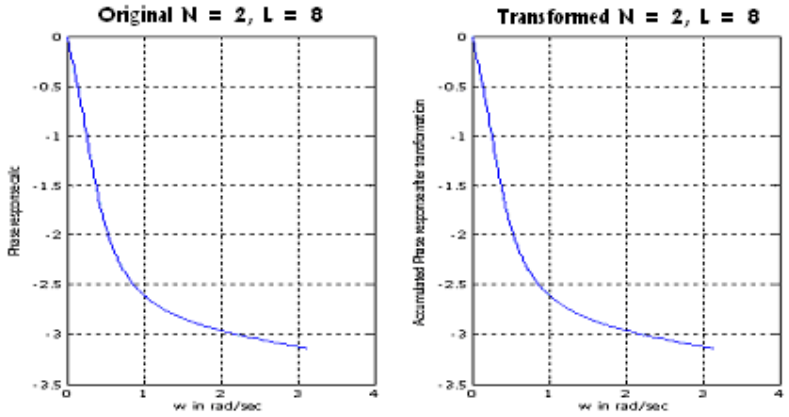


Fig.5.7(b) Phase Responses

Fig. 5.7 (a) Magnitude and (b) phase response of original and transformed filter (N = 2, L = 8, Butterworth Filter)

5.4 *Effect of Coefficient Quantization*

A digital system can be implemented in a number of different structures. Although all structures are functionally equivalent, yet some structures are more sensitive than others to the quantization of coefficients [9, 10]. Computers store numbers in registers that have to be finite in length. Moreover, multiplications and additions generally lead to an increase in the word length as a result of operation. Limitations of the word length change the coefficients and hence frequency response of the system. It also introduces errors due to truncation or overflow of the results inside the system. Due to this the output of digital filters deviates from the designed response. In FIR systems, it affects only the frequency response but in case of IIR systems, it may lead to instability.

Due to hardware limitations, finite number of bits is used to represent the coefficients of a filter [21]. The main effect of this quantization is the displacement of poles and zeros of original transfer function. The process of quantization of numbers can be modeled as shown in Fig.5.8 and the truncation error :

$e_t = Q(x) - x$ for fixed point implementation can be estimated as follows.

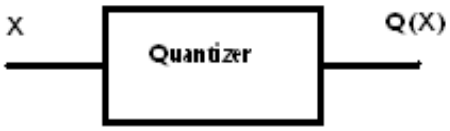


Fig.5.8 The model of a quantizer

If the word length is fixed as (b+1) bits then in fixed point arithmetic using quantization step of 2^{-b}., a positive number of (k+1) bits has a range of truncation error;

$$-\left(2^{-b}-2^{-k}\right)\leq e_t \leq 0.$$

For a negative number, in 2's complement format, the truncation error is always negative and has the same range as given above.

In practice k>>b so 2^{-k} can be neglected as compared to 2^{-b}. In this way, the range of truncation error becomes as:

$$-2^{-b} < e_t \leq 0.$$

For DF-I, DF-II, TDF-I and TDF-II implementation of IIR filter as one single section, quantization of one coefficient of polynomial affects all the poles of the filter. In cases the cumulative effect of quantization of all coefficients may take some of the poles outside the unit circle. In cascade and parallel realizations, the system is decomposed into second order direct form sections. In case of parallel form the effect of quantization on pole-zero pair is limited to the respective first or the second order sections only. Here one second order section realizes one-pair of complex conjugate poles, independent of all other poles. The decomposition of transfer function as first and second order section thus ensures better stability after quantization.

In our technique, the transfer function is decomposed into first order parallel sections. Two first order sections with complex conjugate coefficients are combined to give second order section. The equivalence of transformed filter with the original filter was shown in previous sections assuming infinite precision. The registers to store the data in the system have fixed number of bits so the results are either rounded or truncated. The effect of using finite word length is investigated by quantizing the data values and studying the shift in pole-zero positions. A Matlab program is written to perform this simulation by changing the number of bits used for quantization. The program is given at Appendix-VII. The results are shown in the following Fig.5.9, 5.10, 5.11 and 5.12.

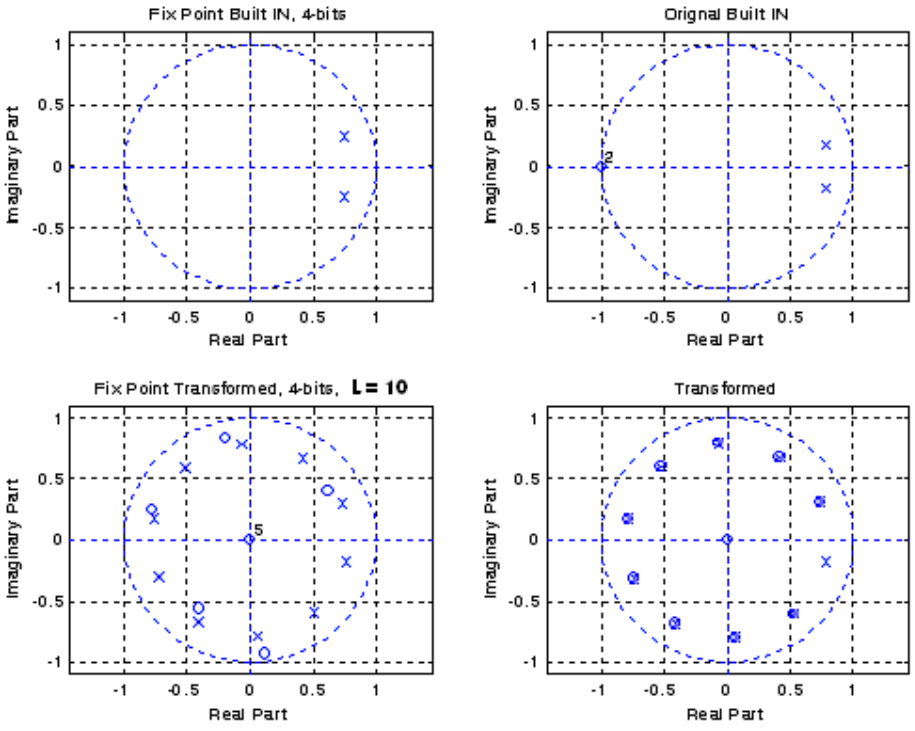


Fig.5.9 Pole-zero plots for 4 bits, L = 10

The figures in the right column represent the pole-zero plots of the original and transformed filters with infinite precision. The figures in the left column represent the pole-zero plots of the original and transformed filters using 4 bits in fixed point

format to represent the coefficients. Comparing the two lower figures we see that the pole-zero location is changed. This results in change in the filter behavior. Thus 4-bit implementation is not recommended.

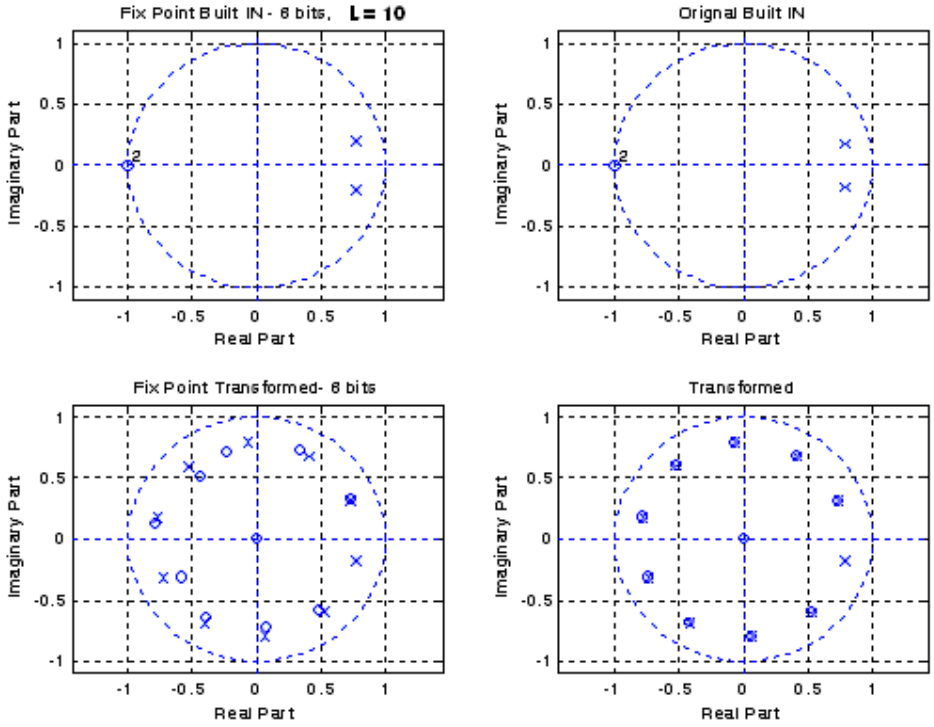


Fig.5.10 Pole-zero plots for 6 bits, L = 10.

Fig.5.10 shows the pole-zero plots for 6-bit implementation. Change in pole-zero patterns is less as compared to 4-bit implementation. However, 6-bit implementation also alters the frequency response of the filter.

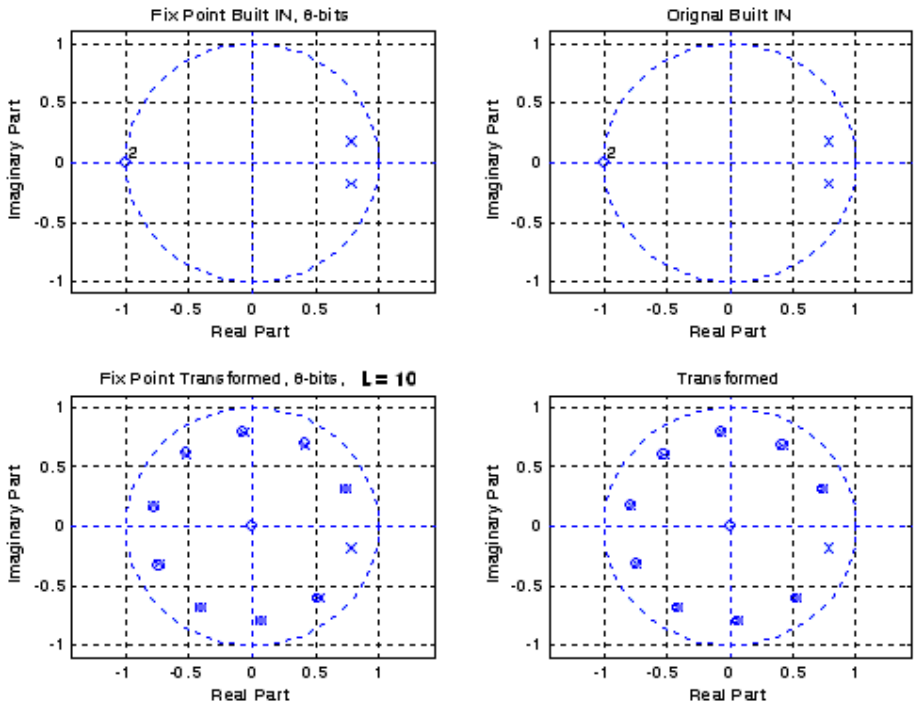


Fig.5.11 Pole-zero plots for 8 bits, L = 10.

Fig.5.11 shows the pole-zero plots for 8-bit implementation in fixed point format. Here we see that the pole-zero pattern is not changed. It means that 8-bit implementation is quite satisfactory.

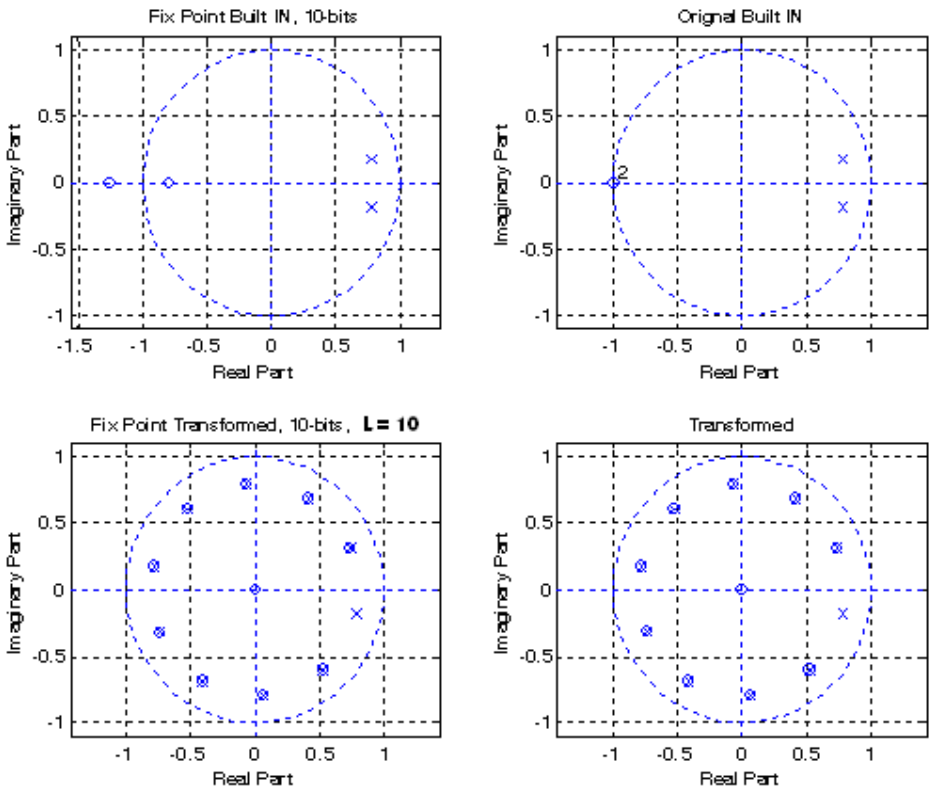


Fig.5.12 Pole-zero plots for 10 bits, L = 10.

Increasing number of bits to 10 does not provide any improvement. This fact is clear from the pole-zero plots of Fig.5.12. The effect of increasing L is also investigated. The pole-zero plots for 8-bit implementation using L = 20 are shown in Fig.5.13.

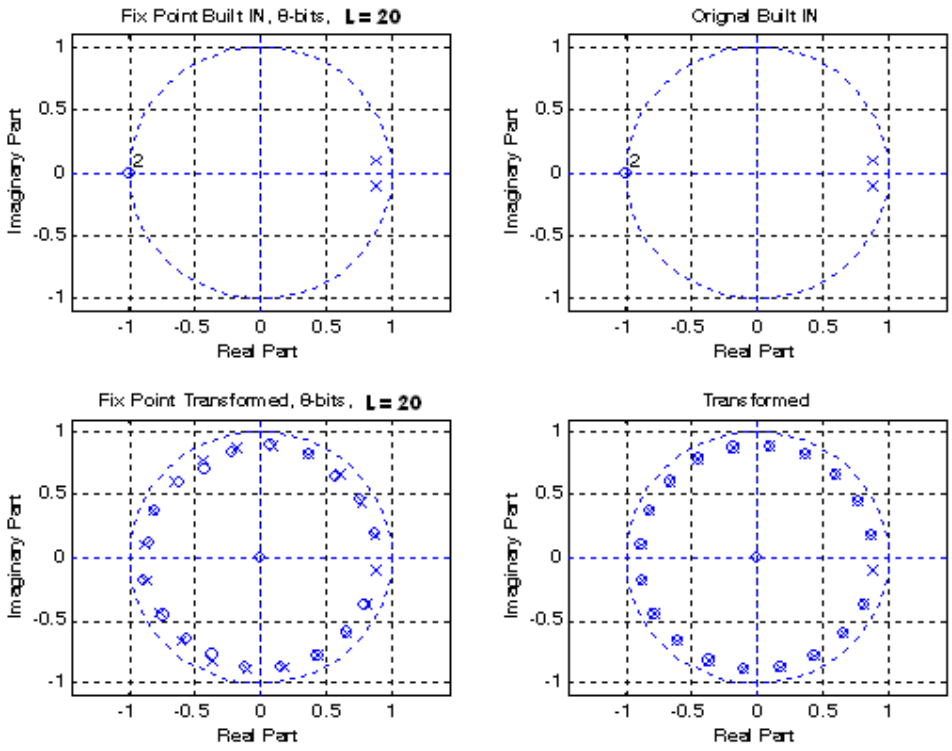


Fig.5.13 Pole-zero plots for 8 bits, L = 20.

Simulation results for 10-bits, 12-bits for L = 20 are also provided for reference.

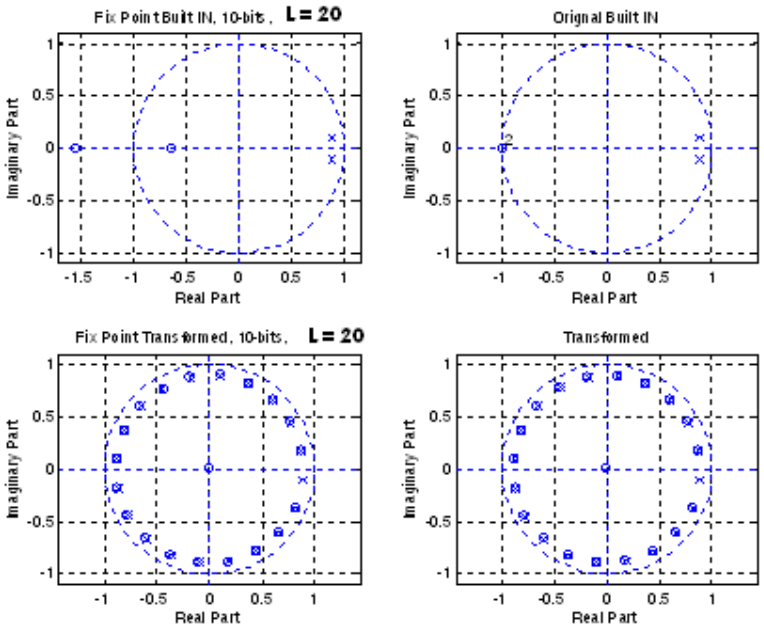


Fig.5.14 Pole-zero plots for 10 bits, L = 20.

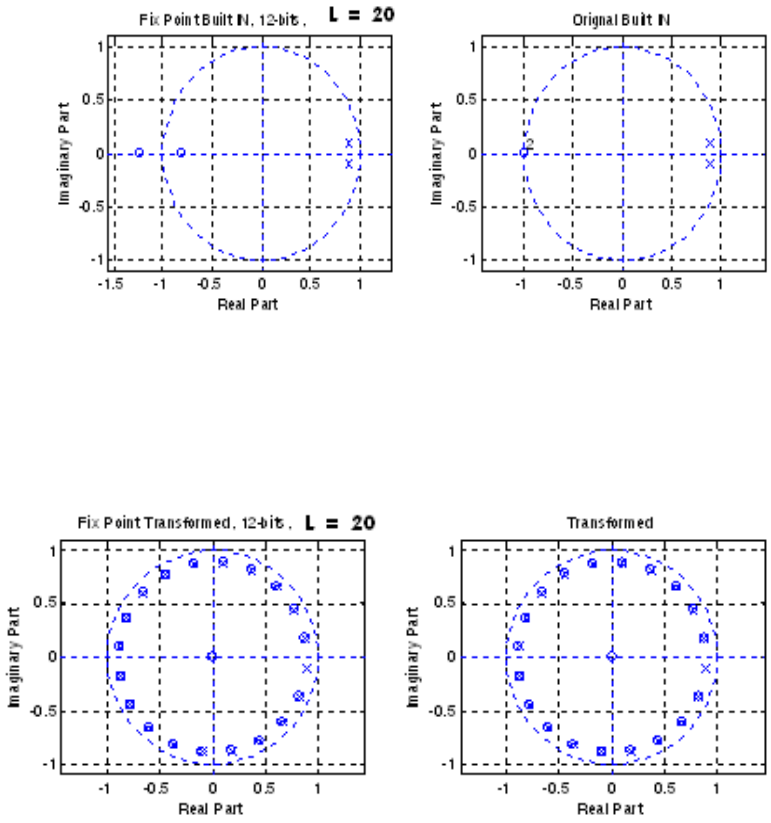


Fig.5.15 Pole-zero plots for 12-bits L = 20.

5.5 Conclusions

The hardware of transformed filter was implemented using Verilog HDL and the filter was synthesized on an FPGA using Spartan-II technology. The study of pole-zero plots of original and transformed filters showed that the transformed filter introduced additional poles and zeros. The pole-zero pairs were at the coincident positions in z-plane. Magnitude and phase responses were unaltered. The filters were checked for coefficient quantization effects using various bits to represent the coefficients. Implementation in fixed point format using 8-bit word length gave excellent results. The simulation results for 10 and 12 bits were also reported.

5.6 References

[1] Ho, H., Szwarc, V., and Kwasniewski, T., "Hardware optimization for a reconfigurable polyphase-FFT design using common sub-expression elimination," Circuits and Systems, 2007. MWSCAS 2007. 50th Midwest Symposium on, 5 – 8 Aug. 2007, Page(s): 650 – 653.

[2] Saramaki, T., Yli-Kaakinen, J., "A Novel Systematic Approach for Synthesizing Multiplication-Free Highly-Selective FIR Half-Band Decimators and Interpolators," Circuits and Systems, 2006, APCCAS 2006, IEEE Asia Pacific Conference on, 4 – 7 Dec 2006, Page(s): 920 – 923.

[3] M. B. Yeary, W. Zhang, J. Q. Trelewicz, Y. Zhai, and B. Mcguire, "Theory and Implementation of a Computationally Efficient Decimation Filter for Power-Aware Embedded Systems," Instrumentation and Measurement, IEEE Transactions on Volume 55, Issue 5, Oct 2006, Page(s): 1839 – 1849.

[4] Tecpanecatl-Xihuitl, J. Luis, Aguilar-Ponce, Ruth M., Ismail, Yasser, Bayoumi, Magdy A., "Efficient multiplierless polyphase FIR filter based on new distributed arithmetic architecture," Signals, Systems and Computers, 2007, ACSSC 2007. Conference Record of the Forty-First Asilomar Conference, 4 – 7 Nov 2007, Page(s): 958 – 962.

[5] Ying Yi, Mark Milward, Sami Khawam, Ioannis Nousias, and Tughrul Arslan, "Automatic Synthesis and Scheduling of Multirate DSP Algorithms," ASP-DAC 2005, IEEE Page(s): 635 – 638.

[6] Michael D. Ciletti, "Advanced Digital Design with the Verilog HDL," Pearson Education 2003.

[7] Shoab Ahmed Khan, “Advanced Digital Design” Video Lectures, CASE, Islamabad, 2008.

[8] Himanshu Bhatnagar, “Advanced Asic Chip Synthesis”, Kluwer Academic Publishers, Second Edition 2002.

[9] Lindberg, Martin and Popp Andreas, “Numerical analysis of finite word length effects in multirate filter system, “ Norchip, 2007, 19 – 20 Nov. 2007, Page(s): 1 – 4.

[10] Artur Krukowski, Richard Charles, Spicer Morling and Izzet Kale, “Quantization effects in the polyphase N-path IIR structure,” Instrumentation and Measurement, Transactions IEEE on, Volume 51, No.6, December 2002, Page(s): 1271 – 1278.

[11] fred harris, “Multirate Signal Processing for Communication Systems”, Prentice Hall, 2004.

[12] P.P. Vaidyanathan, “Multirate Systems and Filter Banks”, Pearson Education, Inc., ch.4, 2004.

[13] Alan V. Oppenheim and Ronald W. Schafer, “Discrete-Time Signal Processing“, Prentice Hall of India, ch.6, 2001.

[14] Emmanuel C. Ifeachor and Barrie W.Jervis, “Digital Signal Processing- A Practical Approach”, Second Edition, Ch.9, 2002, Pearson Education Ltd.

[15] Maurice Bellanger, “Digital Processing of Signals- Theory and Practice”, Second Edition, Ch.10, 1989, John Wiley & Sons.

[16] S.K.Mitra, Digital Signal Processing: A Computer Based Approach, 2nd Edition, McGraw Hill, 2002.

[17] Alan V. Oppenheim, Ronald W. Schafer with John R. Buck, “Discrete-Time Signal Processing,“ Ch.5 and 6, Second Edition, 2000 Pearson Education Asia Pte Ltd. India.

[18] Emmanuel C. Ifeachor and Barrie W. Jervis, “Digital Signal Processing – A Practical Approach”, Second Edition, Pearson Education India, 2002.

[19] MATLAB (R14SP1), Filter Design Toolbox™ 4.3, The MathWorks Inc., 2007.

[20] Maurice Bellanger, “Digital Processing of Signals – Theory and Practice”, Second Edition, John Wiley & Sons Ltd UK, 1989.

[21] A. Krukowski, R.C.S. Morling and I. Kale, “Quantization Effects in the Polyphase N-path IIR Structure,” IEEE Trans. On Instrumentation and Measurement, Vol.51, No.6, December 2002. Page(s). 1271-1278.

…….

CHAPTER- 6

CONCLUSIONS

An efficient architectural transformation was introduced through which sample rate changes and filtering were realized in the same stage using recursive filters. In this technique, an efficient and stable IIR filter was designed using the existing filter design techniques. The filter was decomposed into parallel first order sections. An architectural transformation known as Merged Delay Transformation was introduced that transformed the first order recursive filter into a multirate filter. This transformation converted the transfer function of recursive filters into the form of $H(z^M)$. By arranging down samplers or up samplers with the transformed filter, an efficient architecture of recursive decimator/interpolator was obtained. Transformation equations for second order section were derived by combining a pair of first order sections with complex conjugate coefficients. An optimized architecture for a second order section was derived. An efficient architecture was extracted and implemented for sampling rate changes.

Stability of the transformed filter was investigated through pole-zero plots. It was found that the poles of the system always lie inside unit circle in z-domain that ensures the stability. In the transformed filter, additional poles and zeros equal in number, were introduced at the coincident positions inside the unit circle so magnitude and phase responses were not changed. Several filters were designed and transformed to decimation/Interpolation filters. The magnitude and phase responses of the transformed filters showed close agreement with the original filters. Computational costs in terms of number of multiplies per output sample were compared with FIR polyphase and Conventional IIR structures.

Compared with standard IIR structure, the reductions in computational cost of about 48% and 45% were achieved in decimators and interpolators respectively. As compared to polyphase FIR filters, the reductions in cost were of the order of 82% and 68% for decimators and interpolators respectively.

This architecture requires less hardware and is faster due to parallel implementation. It is expected to find wide use in multirate signal processing.

Efficient hardware implementation on FPGA of higher order recursive decimation /interpolation filters will be explored further. The concept of merged delay transformation will be extended to realize sample rate converters having non-integer interpolation/decimation factors. A parameterized IP core of multirate filters based on merged delay transformation will be implemented. Moreover, low power and reconfigurable architectures will be investigated in the realization of multirate filters.

……..

Appendix – I

Calculation of Output for Second Order Section with Complex Conjugate Coefficients:

A second order transfer function H(z) is given below

$$H(z) = k + \sum_{i=1}^{2} H_i(z) \qquad \text{Eq(A - 1)}$$

where

$$H_1(z) = \frac{r_1}{1 - p_1 z^{-1}} \qquad \text{Eq(A - 2)}$$

$$H_2(z) = \frac{r_2}{1 - p_2 z^{-1}} \qquad \text{Eq(A - 3)}$$

The difference equations from (A-2) and (A-3) can be written as follows.

$$y_1[n] = p_1 y[n-1] + r_1 x[n] \qquad \text{Eq(A - 4)}$$

$$y_2[n] = p_2 y[n-1] + r_2 x[n] \qquad \text{Eq(A - 5)}$$

Assuming r_1 , r_2 and p_1 , p_2 are complex conjugates we can write as follow.

$$r_1 = r_{1r} + jr_{1i} \quad r_2 = r_1^* = r_{1r} - jr_{1i}$$

Eq(A-6)

$$p_1 = p_{1r} + jp_{1i} \quad p_2 = p_1^* = p_{1r} - jp_{1i}$$

Eq(A-7)

From section 2.2, Eq(A-4) and Eq(A-5) can be written in the transformed form as follows.

$$y_1[n] = p_1^M y_1(n-M) + \sum_{k=0}^{M-1} r_1 p_1^k x(n-k) \qquad \text{Eq(A - 8)}$$

$$y_2[n] = p_2^M y_2(n-M) + \sum_{k=0}^{M-1} r_2 p_2^k x(n-k) \qquad \text{Eq(A - 9)}$$

Since complex coefficients are involved so $y_1[n]$ and $y_2[n]$ will in general be complex. Let us represent $y_1[n]$ and $y_2[n]$ as complex quantity.

$$y_1[n] = y_{1R}[n] + jy_{1I}[n] \qquad \text{Eq(A-10)}$$

$$y_2[n] = y_{2R}[n] + jy_{2I}[n] \qquad \text{Eq(A-11)}$$

Expanding the summation and computing the complex coefficients in Eq(A-8, A-9), we find that

$y_{1R}[n] = y_{2R}[n]$
and
$y_{1I}[n] = -y_{2I}[n]$

Hence

$$y_{out}[n] = kx[n] + 2y_{1R}[n] \qquad \text{Eq.(A-12)}$$

$y_{1R}[n]$ is given as follows.

$$y_{1R}[n] = A_{MR}\, y_{1R}[n-M] - A_{MI}\, y_{1I}[n-M] + \sum_{k=0}^{M-1} B_{kR}\, x[n-k] \qquad \text{Eq.(A - 13)}$$

$$y_{1I}[n] = A_{MI}\, y_{1R}[n-M] + A_{MR}\, y_{1I}[n-M] + \sum_{k=0}^{M-1} B_{kI}\, x[n-k] \qquad \text{Eq.(A - 14)}$$

The values of A_{MR}, A_{MI}, B_{kR} and B_{kI} are computed as follows.

$$A_{MR} = \text{real}\ (p_1^M)$$
$$A_{MI} = \text{imag}\ (p_1^M)$$
$$B_{kR} = \text{real}\ (p_1^k r_1)$$
$$B_{kI} = \text{imag}\ (p_1^k r_1) \qquad \text{Eq.(A - 15)}$$

Eq.(A-13) and (A-14) can be solved simultaneously. Taking z-transform of these equations, and shifting the terms containing output y on the left hand side, we can write as follows.

$$y_{1R}(z)[1 - A_{MR}z^{-M}] + A_{MI}z^{-M}y_{1I}(z) = \sum_{k=0}^{M-1} B_{kR}z^{-k}X(z)$$

$$-A_{MI}z^{-M}y_{1R}(z) + [1 - A_{MR}z^{-M}]y_{1I}(z) = \sum_{k=0}^{M-1} B_{kI}z^{-k}X(z)$$

These equations can be put in the matrix form as follows.

$$\begin{bmatrix} 1 - A_{MR}z^{-M} & A_{MI}z^{-M} \\ -A_{MI}z^{-M} & 1 - A_{MR}z^{-M} \end{bmatrix} \begin{bmatrix} y_{1R}(z) \\ y_{1I}(z) \end{bmatrix} = \begin{bmatrix} \sum_{k=0}^{M-1} B_{kR}z^{-k}X(z) \\ \sum_{k=0}^{M-1} B_{kI}z^{-k}X(z) \end{bmatrix} \qquad \text{Eq.(A - 16)}$$

From above, $y_{1R}(z)$ and $y_{1I}(z)$ can be obtained. The value of $y_{1R}(z)$ is:

$$Y_{1R}(z) = \frac{X(z)[(1 - A_{MR}z^{-M})\sum_{k=0}^{M-1} B_{kR}z^{-k} - A_{MI}z^{-M}\sum_{k=0}^{M-1} B_{kI}z^{-k}]}{1 - 2A_{MR}z^{-M} + (A_{MR}^2 + A_{MI}^2)z^{-2M}} \qquad \text{Eq.(A - 17)}$$

The transfer function $H_{1R}(z) = Y_{1R}(z)/X(z)$ can be obtained as follows.

$$H_{1R}(z)=\frac{(1-A_{MR}z^{-M})\sum_{k=0}^{M-1}z^{-k}B_{kR} - A_{MI}z^{-M}\sum_{k=0}^{M-1}z^{-k}B_{kI}}{1-2A_{MR}z^{-M}+A_{MR}^{2}z^{-2M}+A_{MI}^{2}z^{-2M}}$$

Eq(A-18)

Similarly $H_{1I}(z)$, $H_{2R}(z)$ and $H_{2I}(z)$ can be obtained but they are not required.

.....

Appendix-II

```
%Matlab Program A
%CheckFiltergeneral.m
%written by Prof. Umar Farooq, EED UET Taxila, November 4, 2007.
%Program to compare the outputs from Original Filter and Transformed Filter
using first order sections only
%For a fixed value of Decimation Factor M, it can design Four Types of
%Filters. It generates output for all values of filter order ranging from 1
%to order.
%Decimation Factor M is to be given any integer value
%Order means the filter order that is to be supplied as input as integer
%value
%type means the filter Type.
%type=1 will design Chebyshev-I filter,
%type=2 will design Butterworth Filter,
%type=3 will design elliptical filter and
%type=4 will design Chebyshev-II Filter.
close all
clear all
deci_fac=10;
M=deci_fac;
order=6;
type=3;
for n=1:order

   switch type
```

```
    case 1
%         title('Cheby 1')
        s='Cheby 1   order = ';
      [b,a]=cheby1(n,0.01,1/M);
       case 2
%           title('Butterworth')
        s='Butterworth   order = ';
      [b,a]=butter(n,1/M);
       case 3
%           title('Elliptical')
          s='Elliptical   order = ';
      [b,a] = ellip(n,0.001,80,1/M)
        case 4
%            title('Cheby 2')
           s='Cheby 2    order = ';
      [b,a]=cheby2(n,0.01,1/M);


    end
   figure(n);
   s=strcat(s,mat2str(n),', M = ',mat2str(M));
   [h,w]=freqz(b,a);
   [r,p,k]=residuez(b,a);
   x=rand(100,1);


   ydp=0;
   hd=0;
```

```
for m=1:n;
    bp=[1]; ap=[1];
    for c=1:M-1;
        bp=[bp, p(m)^c];
        ap=[ap, 0];
    end
    ap=[ap, -p(m)^M];
    ydp=ydp+filter(r(m)*bp,ap,x);
    [hdp,wdp]=freqz(r(m)*bp,ap);
    hd=hd+hdp;
end
hd=hd+k;
    yd=k*x+real(ydp);

x1=x';
y=filter(b, a, x1);
x=[zeros(1,M),x1];
ydirect=[zeros(1,M),zeros(1,length(x1))];
for n=M+1:length(x);
    ytemp=0;
    for m=1:(length(b)-1);
        ytemp=ytemp-a(m+1)*ydirect(n-m)+b(m+1)*x(n-m);
    end
    ydirect(n)= b(1)*x(n)+ytemp;
end
ydirect=ydirect(M+1:length(ydirect));
```

```
    yd=yd';
  diff = yd(1:M:end)- ydirect(1:M:end);

  stem(diff);title(s);grid on;ylabel('Difference between o/p');xlabel('samples');
end
```

……..

Appendix-III

```
%Program B to check the Real and Imaginary values in a second order section
%Program written by Prof. Umar Farooq, EED UET Taxila. 6-11-2007
%Decimation factor is Two
%
close all
clear all
deci_fac=2;
M=deci_fac;
[b, a]= butter(2, 1/M);
[r,p,k]=residuez(b,a);
x1=rand(1,10);
deci_fac=2;
% % % % % % %
% % % % % % % % zeros are appended on left to avoid negative indexing so
first output
% % % % % % % % sample y(0) is shifted to deci_fac+1
% % % % % % %
x=[zeros(1,deci_fac),x1];

y1I=[zeros(1,deci_fac),zeros(1,length(x1))];
y1R=[zeros(1,deci_fac),zeros(1,length(x1))];
y2I=[zeros(1,deci_fac),zeros(1,length(x1))];
y2R=[zeros(1,deci_fac),zeros(1,length(x1))];
p1r=real(p(1));
```

```
p1i=imag(p(1));

r1r=real(r(1));

r1i=imag(r(1));

A=p1r^2 - p1i^2;

B=2*p1r*p1i;


C=r1r;

D=r1i;


E=r1r*p1r - r1i*p1i;

F=r1i*p1r + r1r*p1i;

 imagp1(1)=D*x(1) + F*0;

 realp1(1)=C*x(1) + D*0;


    y1R(1)=A*0 - B*0 + realp1(1);

    y1I(1)=A*0 + B*0 + imagp1(1);

    y2R(1)=A*0 + B*0 + realp1(1);

    y2I(1)=A*0 - B*0 - imagp1(1);


for n=deci_fac+1:length(x)

    imagp1(n)=D*x(n) + F*x(n-1);

    realp1(n)=C*x(n) + D*x(n-1);

    y1R(n)=A*y1R(n-2) - B*y1I(n-2) + realp1(n);

    y1I(n)=A*y1I(n-2) + B*y1R(n-2) + imagp1(n);

    y2R(n)=A*y2R(n-2) + B*y2I(n-2) + realp1(n);

    y2I(n)=A*y2I(n-2) - B*y2R(n-2) - imagp1(n);
```

```
end
y1I=y1I(deci_fac+1:length(y1I))
y1R=y1R(deci_fac+1:length(y1R))
y2I=y2I(deci_fac+1:length(y2I))
y2R=y2R(deci_fac+1:length(y2R))
```

.....

Appendix-IV

```
%Program C to check the equivalence of calculation of output from real parts
%only in a pair of second order section with complex cnjugate coefficients
%Second Order, Decimation factor M = 4
%Program Written by Prof. Umar Farooq, EED UET Taxila: Dated 6-11-2007
%
clc;
clear;
close all;
deci_fac=4;
M=deci_fac;
 [b,a]=cheby1(2,.01,1/M);
%[b,a]=butter(2,1/M);

figure(1); freqz(b,a);

[r,p,k]=residuez(b,a);

x=rand(1,100);

y=filter(b,a,x);

    A=real(p(1)^4);
    B=imag(p(1)^4);
    C=real(p(2)^4);
```

```
D=imag(p(2)^4);
y1I=zeros(1,length(x));
y1R=zeros(1,length(x));
y2I=zeros(1,length(x));
y2R=zeros(1,length(x));
yout=zeros(1,length(x));
y1R(1)= real(r(1)*x(1));y1I(1)= imag(r(1)*x(1));
y2R(1)= real(r(2)*x(1));y2I(1)= imag(r(2)*x(1));


for n=deci_fac+1:4:length(x)
    y1R(n)=A*y1R(n-4)- B*y1I(n-4)+ real(r(1)*x(n)+ r(1)*p(1)*x(n-
1)+r(1)*(p(1)^2)*x(n-2)+...
        r(1)*(p(1)^3)*x(n-3));
    y1I(n)=A*y1I(n-4) + B*y1R(n-4)+ imag(r(1)*x(n)+ r(1)*p(1)*x(n-
1)+r(1)*(p(1)^2)*x(n-2)+...
        r(1)*(p(1)^3)*x(n-3));

    y2R(n)=C*y2R(n-4) - D*y2I(n-4)+ real(r(2)*x(n)+ r(2)*p(2)*x(n-
1)+r(2)*(p(2)^2)*x(n-2)+...
        r(2)*(p(2)^3)*x(n-3));

    y2I(n)=C*y2I(n-4) + D*y2R(n-4)+ imag(r(2)*x(n)+ r(2)*p(2)*x(n-
1)+r(2)*(p(2)^2)*x(n-2)+...
        r(2)*(p(2)^3)*x(n-3));
```

```
    end

    yout= y2R + y1R + k*x;



diff=y(1:4:end)-yout(1:4:end);



figure (2)
stem(y(1:4:end));grid on;ylabel('Calculated outputs (circles), decimator output (cross)');xlabel('samples');
hold on
plot(yout(1:4:end),'xr');
hold off
figure (3)
stem(diff);grid on;ylabel('Difference between direct and decimated outputs');xlabel('samples'); title('Second order using real outputs, M = 4')
```

……

Appendix-V

```
%Program D to Compare the outputs from Original and Transformed Filters
%Program Written by Prof. Umar Farooq, EED, UET Taxila, dated:6-11-2007
%Second Order Butterworth Filter is designed and original outputs are
%calculated
%Output of transformed Filter is calculated using only the real output of first
%order section
%
clear;
close all;
deci_fac=10;
M=deci_fac;
%[b,a]=cheby1(2,.01,1/M);
[b,a]=butter(2,1/M);

figure(1); freqz(b,a);

[r,p,k]=residuez(b,a);

x=rand(1,100);

y=filter(b,a,x);



   A=real(p(1)^10);
```

```
B=imag(p(1)^10);
C=real(p(2)^10);
D=imag(p(2)^10);
y1I=zeros(1,length(x));
y1R=zeros(1,length(x));
y2I=zeros(1,length(x));
y2R=zeros(1,length(x));
yout=zeros(1,length(x));
y1R(1)= real(r(1)*x(1));y1I(1)= imag(r(1)*x(1));
y2R(1)= real(r(2)*x(1));y2I(1)= imag(r(2)*x(1));

for n=deci_fac+1:10:length(x)
    y1R(n)=A*y1R(n-10)- B*y1I(n-10)+ real(r(1)*x(n)+ r(1)*p(1)*x(n-
1)+r(1)*(p(1)^2)*x(n-2)+...
        r(1)*(p(1)^3)*x(n-3)+ r(1)*(p(1)^4)*x(n-4)+r(1)*(p(1)^5)*x(n-
5)+r(1)*(p(1)^6)*x(n-6)+r(1)*(p(1)^7)*x(n-7)+...
        r(1)*(p(1)^8)*x(n-8)+r(1)*(p(1)^9)*x(n-9));
    y1I(n)=A*y1I(n-10) + B*y1R(n-10)+ imag(r(1)*x(n)+ r(1)*p(1)*x(n-
1)+r(1)*(p(1)^2)*x(n-2)+...
        r(1)*(p(1)^3)*x(n-3)+ r(1)*(p(1)^4)*x(n-4)+r(1)*(p(1)^5)*x(n-
5)+r(1)*(p(1)^6)*x(n-6)+r(1)*(p(1)^7)*x(n-7)+...
        r(1)*(p(1)^8)*x(n-8)+r(1)*(p(1)^9)*x(n-9));

    y2R(n)=C*y2R(n-10) - D*y2I(n-10)+ real(r(2)*x(n)+ r(2)*p(2)*x(n-
1)+r(2)*(p(2)^2)*x(n-2)+...
```

```
        r(2)*(p(2)^3)*x(n-3)+                r(2)*(p(2)^4)*x(n-4)+r(2)*(p(2)^5)*x(n-
5)+r(2)*(p(2)^6)*x(n-6)+r(2)*(p(2)^7)*x(n-7)+...
        r(2)*(p(2)^8)*x(n-8)+r(2)*(p(2)^9)*x(n-9));

    y2I(n)=C*y2I(n-10)  +  D*y2R(n-10)+  imag(r(2)*x(n)+  r(2)*p(2)*x(n-
1)+r(2)*(p(2)^2)*x(n-2)+...
        r(2)*(p(2)^3)*x(n-3)+                r(2)*(p(2)^4)*x(n-4)+r(2)*(p(2)^5)*x(n-
5)+r(2)*(p(2)^6)*x(n-6)+r(2)*(p(2)^7)*x(n-7)+...
        r(2)*(p(2)^8)*x(n-8)+r(2)*(p(2)^9)*x(n-9));

  end

  yout= y2R + y1R + k*x;

diff=yout-y;
figure (2)
stem(y(1:10:end));grid  on;ylabel('Calculated  outputs  (circles),  decimator  output
(cross)');xlabel('samples');
hold on
plot(yout(1:10:end),'xr');
hold off
figure (3)
stem(diff(1:10:end));grid  on;ylabel('Difference  between  direct  and  decimated
outputs');xlabel('samples');
```

Appendix-VI

```
%Program E
%written by Prof. Umar Farooq, EED UET Taxila, November 6, 2007.
%Program to compare the Magnitude and phase reponses of Original Filter and
Transformed Filter
%For a fixed value of Decimation Factor M, it can design Four Types of
%Filters. It generates output for all values of filter order ranging from 1
%to order.
%Decimation Factor M is to be given any integer value
%Order means the filter order that is to be supplied as input as integer
%value
%type means the filter Type.
%type=1 will design Chebyshev-I filter,
%type=2 will design Butterworth Filter,
%type=3 will design elliptical filter and
%type=4 will design Chebyshev-II Filter.
close all
clear all
deci_fac=4;
M=deci_fac;
order=1;
type=2;
for n=1:order

   switch type
```

```
    case 1
%         title('Cheby 1')
        s='Cheby 1   order = ';
      [b,a]=cheby1(n,0.01,1/M);
       case 2
%           title('Butterworth')
        s='Butterworth   order = ';
      [b,a]=butter(n,1/M);
       case 3
%           title('Elliptical')
          s='Elliptical   order = ';
      [b,a] = ellip(n,0.001,80,1/M)
        case 4
%            title('Cheby 2')
           s='Cheby 2    order = ';
      [b,a]=cheby2(n,0.01,1/M);


    end
   figure(n);
   s=strcat(s,mat2str(n),', M = ',mat2str(M));
   [h,w]=freqz(b,a);
  subplot(2,2,1);   plot(w,abs(h));   title(s);   grid   on;   ylabel('Mag   response
Original');xlabel('w in rad/sec');
  subplot(2,2,2);   plot(w,angle(h));   title(s);   grid   on;   ylabel('Phase   response
Original');xlabel('w in rad/sec');
   [r,p,k]=residuez(b,a);
```

```
x=rand(100,1);


ydp=0;
hd=0;
for m=1:n;
   bp=[1]; ap=[1];
   for c=1:M-1;
      bp=[bp, p(m)^c];
      ap=[ap, 0];
   end
   ap=[ap, -p(m)^M];
   ydp=ydp+filter(r(m)*bp,ap,x);
   [hdp,wdp]=freqz(r(m)*bp,ap);
   hd=hd+hdp;
end
hd=hd+k;
   yd=k*x+real(ydp);
subplot(223);  plot(wdp,abs(hd));title(s);grid  on;ylabel('Mag  response  after
Transformation');xlabel('w in rad/sec');
subplot(224);  plot(wdp,angle(hd));title(s);grid  on;ylabel('Phase  response  after
Transformation');xlabel('w in rad/sec');
x1=x';
y=filter(b, a, x1);
x=[zeros(1,M),x1];
ydirect=[zeros(1,M),zeros(1,length(x1))];
for n=M+1:length(x);
```

```
        ytemp=0;
        for m=1:(length(b)-1);
            ytemp=ytemp-a(m+1)*ydirect(n-m)+b(m+1)*x(n-m);
        end
        ydirect(n)= b(1)*x(n)+ytemp;
    end
    ydirect=ydirect(M+1:length(ydirect));
        yd=yd';
    diff = yd(1:M:end)- ydirect(1:M:end);
    %subplot(325);stem(yd(1:M:end));title(s);grid on;ylabel('Output
calculated');xlabel('samples');
    %hold on
    %subplot(325);plot(ydirect(1:M:end),'xr');title(s);grid on; ylabel('Output by
Algo');xlabel('samples');
    %hold off
    %subplot(326);
    %stem(diff);title(s);grid on;ylabel('Difference between o/p');xlabel('samples');
end
```

……

Appendix-VII

```
%Program for checking the effect of Coefficient Quantization on poles and zeros
of decimated IIR filter
%Effect of Coefficient Quantization on magnitude and phase responses of
original and transformed
%filters
%Program written by Prof. Umar Farooq, EED, UET Taxila, dated: 8-11-2007
clc;
clear all;
close all;
deci_fac=20;%%Decimation factor
M=deci_fac;
factor=8;%%%%%%Number of bits for quatization
order= 2;%%%Order of the filter
n= order;
[b,a]=butter(n,1/M);
 [r,p,k]=residuez(b,a);


  for m=1:n;
          bp=[1]; ap=[1];
       for c=1:M-1;
          bp=[bp, p(m)^c];
          ap=[ap, 0];
       end
          bp=[r(m)*bp, 0];
```

```
        ap=[ap, -p(m)^M];
 [h,w]=freqz(b,a);
 if(m>1)
 temp=hd;
 else
    temp=0;
 end
  [hd,wd]=freqz(bp,ap);
  save=hd;
  hd=hd+temp;
  af=fix(a*2^factor);
  bf=fix(b*2^factor);
  apf=fix(ap*2^factor);
  bpf=fix(bp*2^factor);
   [hdf,wdf]=freqz(bpf,apf);
   [hf,wf]=freqz(bf,af);
   kf=fix(k*2^factor);
  if m==1
    hd=hd+k;
    hdf=hdf+kf;
    hf=hf+kf;
  else
  end
if (m==n   )
  figure,
    subplot(2,4,1); plot(w,abs(h));grid on;
```

```
    ylabel('Built In Amplitude')
    subplot(2,4,2); plot(wf,abs(hf));grid on;
    ylabel('FIxed Point Built In Amplitude Response')
    subplot(2,4,3); plot(wd,abs(hd));grid on;
    ylabel('Transformed Amplitude Response ')
    subplot(2,4,4); plot(wdf,abs(hdf));grid on;
    ylabel('Fixed Point Transformed Amplitude Response')
        subplot(2,4,5); plot(w,angle(h));grid on;
    ylabel('Built In Phase')
    subplot(2,4,6); plot(wf,angle(hf));grid on;
    ylabel('FIxed Point Built In Phase Response')
    subplot(2,4,7); plot(wd,angle(hd));grid on;
    ylabel('Transformed Phase Response ')
    subplot(2,4,8); plot(wdf,angle(hdf));grid on;
    ylabel('Fixed Point Transformed Phase Response')
      figure,
  subplot(2,2,1),zplane(bf,af);grid on;
  title('Fix Point Built IN');
  subplot(2,2,2),zplane(b,a);grid on;
  title('Orignal Built IN');
  subplot(2,2,4),zplane(bp,ap);grid on;
  title('Transformed');
  subplot(2,2,3),zplane(bpf,apf);grid on;
  title('Fix Point Transformed');
end
        zd=roots(bp);
```

```
pd=roots(ap);
zk=roots(b);
pk=roots(a);
end
```

…….....