DESIGN AND IMPLEMENTATION OF EFFICIENT STRUCTURES FOR DIGITAL INTERPOLATORS

A THESIS

SUBMITTED IN FULFILLMENT OF THE REQUIREMENT

FOR THE AWARD OF DEGREE OF

DOCTOR OF PHILOSOPHY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

BY

RAJEEV RATAN

SUPERVISORS

DR. SANJAY SHARMA

(PROFESSOR)

DR. AMIT KUMAR KOHLI

(ASSOCIATE PROFESSOR)

ELECTRONICS AND COMMUNICATION ENGINEERING DEPARTMENT

THAPAR UNIVERSITY, PATIALA-147004 (INDIA)

2013

CERTIFICATE

I, **Rajeev Ratan,** hereby declare that the thesis entitled, **"Design and Implementation of Efficient Structures for Digital Interpolators,"** submitted to Thapar University, Patiala, in partial fulfillment of the requirement for the award of the Degree of **Doctor of Philosophy** in **the Electronics and Communication Engineering** is a record of original and independent research work done by me during 2009-2013. This thesis has been conducted under the supervision and guidance of **Dr. Sanjay Sharma**, Professor, Electronics and Communication Engineering Department, Thapar University, and **Dr. Amit Kumar Kohli**, Assistant Professor, Electronics and Communication Engineering Department, Thapar University. It has not formed the basis for the award of any Degree/Diploma/Associate-ship/Fellowship or other similar title to any candidate of any university.

Rajeev Ratan

(Signature of Candidate) Date: 03/01/2014

This is to certify that above statement made by the candidate is correct to the best of my knowledge.

3/1/2014

Dr. Sanjay Sharma
Professor (Supervisor)
Electronics and Communication Engg. Deptt.
Thapar University, Patiala.

03/01/2014

Dr. Amit Kumar Kohli
Assistant Professor (Supervisor)
Electronics and Communication Engg. Deptt.
Thapar University, Patiala.

ii

ABSTRACT

To fulfill the ever increasing demand of modern electronic devices operating at the different sampling rates, the interest has touched its zenith in the up-sampling based discrete-time signal processing techniques, which can be incorporated by using the efficient digital interpolators. In this research work, the digital interpolators have been implemented efficiently and effectively using the field programmable gate arrays (FPGA) technology. It has also been demonstrated that how performance of the digital interpolators can be improved by using different techniques at the different sampling rates. First, the cascaded integrator comb (CIC) filters with compensation have been proposed using FPGA, and the results have been investigated for the different wireless standards. These results demonstrate that the logic utilization for global system for the mobile communication (GSM), wideband code division multiple access (WCDMA) and worldwide interoperability for the microwave access (WiMax) is 93%, 43% and 51% respectively. The worst case set up time for GSM, WCDMA and WiMax is 20.89 ns, 20.58 ns and 7.50 ns respectively. The worst case clock-to-output time for GSM, WCDMA and WiMax is 6.66 ns, 7.01 ns and 6.79 ns respectively.

From the acquired results, it is evident that the CIC filters are efficient for the low-cost applications because the multipliers are not required in its implementation. Due to the absence of multipliers, they exhibit relatively fast response as compared to the conventional technology. However, the pass-band droop present in the CIC filters confine the scope of its practical applications. By employing compensation and multi-stage techniques, the response of CIC filter in the pass-band can be significantly improved, but at the cost of escalated hardware requirement and the computational complexity. For comparison purpose, the CIC filters have been implemented with and without compensation using the FPGA technology. By employing compensation techniques, the

response of CIC filter in the pass-band is found to be significantly improved, as the normalized frequency reduces from 0.47 π rad./sample to 0.12 π rad./sample, but at the cost of increased hardware requirement as well as computational burden. The results show that the logic utilization increases to 63% in the case of CIC structures with compensation, while it is 29% for the CIC structures without compensation. The results clearly depict that the proposed digital interpolator structures can be utilized efficiently in the designing and development of the very large scale integration of the modern digital communication systems.

However in digital communication systems, the receiver clock may have time-varying offset, which causes inter-symbol interference at the receiver output. One way for compensating this error is to fractionally delay the signal. Therefore to obtain the fractional delay, the Farrow filter structures are extensively explored in the literature and are widely used in practice. Subsequently, the design and analysis of interpolation filters using the Lagrange polynomial based Farrow filtering structures have been carried out. The presented Farrow filter structures have been designed using the Altera DSP builder advanced blockset. It may be observed from the presented results that even though the cubic Lagrange interpolation method provides somewhat less smooth interpolated signal as compared to the quadratic Lagrange interpolation method, but the ripple level reduces from 60 dB to approximately 25 dB, and the computational need also alleviates by 42% in case of multipliers and 18% in case of additions in the cubic Lagrange interpolation approach. Therefore, the Farrow filtering structures for the cubic Lagrange interpolation are found to be optimum for the designing of interpolation filters. However, the simulation results have been compiled for the single-stage interpolation filters.

It is noteworthy that the single-stage filters are efficient for the lower order interpolation factors, but for the high rate change (which is required in the modern digital

communication systems like WCDMA, WiMax), the single-stage implementation does not provide sufficiently effective response. Therefore for these applications under similar conditions, the multi-stage implementation of the interpolator may be preferred. Further, the efficient implementation of digital interpolation systems for the up-sampling of quadrature amplitude modulation (QAM)/quadrature phase shift keying (QPSK) based multi-stage digital interpolators have also been proposed. Here, the total output power improves from -25.19 dBm to -22.96 dBm, and the peak to average power improves from 6.166 dB to 6.106 dB in the QAM based double-stage interpolation filter. The total output power improves from -28.09 dBm to -25.84 dBm, and the peak to average power improves from 3.181 dB to 3.158 dB in the QPSK based double-stage interpolation filter. But, the multi-stage implementation in turn increases the computational as well as implementation complexity of the overall communication system. However, this complexity can be reduced considerably by using the half-band filters. For channel bandwidths of 20 MHz, 15 MHz, 10 MHz, 7.5 MHz and 5 MHz, the three-stage filtering without the half-band filters can be realized using 100%, 78%, 49%, 30% and 28% of multipliers respectively, which are essential for single-stage filtering. At the same time, the three-stages filtering with the half-band filters can be realized using 100%, 68%, 36%, 21% and 18% of multipliers respectively, which are essential for single-stage filtering. It is evident from the results that the half-band filters are beneficial in the designing of efficient multi-stage digital interpolators from the implementation perspective.

In the aforementioned research work, we have explored and simulated the CIC filters with compensation on FPGA, the cubic Lagrange polynomial based Farrow filtering structures, the conventional multi-stage filters and the half-band filters for interpolation in the emerging field of communication engineering for the different wireless standards as well as applications.

ACKNOWLEDGEMENT

First and foremost, I am beholden to the Almighty and I bow before Him for his umpteen blessings and bestowing in me the grit and confidence to carry out this research work.

I extend my thanks to our Hon'ble Director, **Dr. K. K. Raina** and **Dr. P. K. Bajpai**, Distinguished Professor and Dean (Research and Sponsored Projects) for giving me this opportunity to undertake the Ph.D.

I would like to put on record my heartfelt and sincere gratitude to my research guide **Dr. Sanjay Sharma**, Professor, Electronics and Communication Engineering Department, Thapar University, Patiala and **Dr. Amit Kumar Kohli**, Assistant Professor, Electronics and Communication Engineering Department, Thapar University, Patiala. I feel bound to be grateful to both my guiding souls for their valuable and constant support, advice, and encouragement.

I am honor bound and profoundly thankful to the doctoral committee members **Dr. Rajesh Khanna**, Professor and Head of Department, Electronics and Communication Engineering Department, Thapar University, Patiala, **Dr. Kulbir Singh**, Associate Professor, Electronics and Communication Engineering Department, Thapar University, Patiala, and **Dr. M. K. Sharma**, Associate Professor, School of Mathematics and Computer Applications for their consistent help.

I also thank the management of MVN University, Palwal, Haryana, India for countenancing me to pursue Ph.D. at Thapar University, Patiala. I would like to extend special thanks to my colleagues and friends **Mr. Deepak Batra, Mr. I. C. Bhardwaj, Mr. Dharmvir Jain,** and **Dr. Shyam Akashe** for their valuable suggestions, support, and encouragement.

It is great privilege to express my profound thankfulness, deep love, and fondness to my wife **Anju** and my children **Garima** and **Vishant**, who stood like a rock in my

difficult times. Their love, patience, persistent encouragement, and good virtuous understanding enabled me to complete the research work successfully.

Finally, I will remain obligated to and filled with appreciation towards my parents for their love, affection, sacrifices, perseverance, and prayers all through my life.

Rajeev Ratan

TABLE OF CONTENTS

CHAPTER 2

EFFECTS OF COMPENSATION AND ARBITRARY SAMPLING IN INTERPOLATORS FOR DIFFERENT WIRELESS STANDARDS ON FPGA PLATFORM

CHAPTER 3

DESIGNING OF EFFICIENT FARROW FILTERING STRUCTURE BASED INTERPOLATORS USING LAGRANGE POLYNOMIAL ON FPGA TECHNOLOGY

CHAPTER 4

PERFORMANCE COMPARISON OF QUADRATURE AMPLITUDE MODULATION (QAM)/ QUADRATURE PHASE SHIFT KEYING (QPSK) BASED SINGLE AND DOUBLE-STAGE DIGITAL INTERPOLATORS

CHAPTER 6

CONCLUDING REMARKS AND FUTURE SCOPE

LIST OF FIGURES

LIST OF TABLES

ACRONYMS AND ABBREVIATIONS

A/D	:	Analog to Digital
ADC	:	Analog to Digital Converter
ADS	:	Advanced Design System
AM	:	Amplitude Modulation
ASIC	:	Application Specific Integrated Circuits
ASK	:	Amplitude Shift Keying
BER	:	Bit Error Rate
BPSK	:	Binary Phase Shift Keying
CDMA	:	Code Division Multiple Access
CIC	:	Cascaded Integrator Comb
D/A	:	Digital to Analog
DAC	:	Digital to Analog Converter
DDC	:	Digital Down Converter
DEM	:	Dynamic Element Matching
DF	:	Direct Form
DFT	:	Discrete Fourier Transform
DQPSK	:	Differential Quadrature Phase Shift Keying
DSP	:	Digital Signal Processing
DUC	:	Digital Up-converter
DVD	:	Digital Versatile Disk
FFD	:	Fixed Fractional Delay
FIR	:	Finite Impulse Response
FM	:	Frequency Modulation
FPGA	:	Field-Programmable Gate Array
FSK	:	Frequency Shift Keying

GSM	:	Global System for Mobile Communication
HSDPA	:	High Speed Downlink Packet Access
I/Q	:	In-Phase and Quadrature
IC	:	Integrated Circuits
IF	:	Intermediate Frequency
IIR	:	Infinite Impulse Response
ISI	:	Inter-symbol Interference
LTI	:	Linear Time Invariant
LUT	:	Look-Up Table
MDT	:	Merged Delay Transformation
ML	:	Maximum Likelihood
MMSE	:	Minimum Mean Square Error
MSE	:	Mean Square Error
QAM	:	Quadrature Amplitude Modulation
QPSK	:	Quadrature Phase Shift Keying
PCM	:	Pulse Code Modulation
PFS	:	Poly-Phase Farrow Structure
PLL	:	Phase Locked loop
PR	:	Perfect Reconstruction
PSK	:	Phase Shift Keying
RF	:	Radio Frequency
SDR	:	Software Defined Radio
SER	:	Symbol Error Rate
SNR	:	Signal to Noise Ratio
SRC	:	Sample Rate Conversion
SVD	:	Singular Value Decomposition
TDF	:	Transposed Direct Form

TDM	:	Time Division Multiplexing
UMTS	:	Universal Mobile Telecommunications System
VFD	:	Variable Fractional Delay
VLSI	:	Very Large Scale Integration
WCDMA	:	Wideband Code Division Multiple Access
WiMax	:	Worldwide Interoperability for Microwave Access
Wi-Fi	:	Wireless Fidelity

CHAPTER 1

INTRODUCTION BASED ON LITERATURE REVIEW

1.1 INTRODUCTION

In the present epoch, almost all the signal processing problems are resolved in the digital domain because of the availability of prominent very large scale integrated (VLSI) circuits. These circuits permit us to carry out complex processes in the real time, without the well-known restraints of the analog implementations. The operations like source signal filtering, shaping, mixing etc. are carried out in the digital domain, and only the final result is transformed back to analog. A very significant and fundamental operation in the discrete time signal processing is sampling. The discrete time signals are frequently obtained from the continuous time signals by the conventional sampling procedure based on the Nyquist criteria. This is mathematically modeled as the assessment of a function of real variable at the discrete time instants. Physically, it is a more intricate process, which might be modeled as convolution of the sampled signal by a narrow pulse.

In numerous applications, it is occasionally essential to transform a signal from one sample rate to another sample rate. An input can be down-sampled or up-sampled before processing to facilitate the computational load, or occasionally two units have to be linked, whose particular sample rates do not match. To subjugate the degradation caused by analog to digital (A/D) and digital to analog (D/A) conversion, all the processing blocks must have digital interfaces. In the modern digital communication systems, a wide variety of sample rates are used depending upon the available bandwidth of the channel, the requisite quality and data rate of the interfaces. The unification of all these systems is however trouble-free, if an appropriate sample rate converter is used at each interface in the communications systems [1], [2].

However, the multi-rate signal processing is a key tool used in these signal processing applications. The digital filters are a vital part of the multi-rate signal processing. The digital filters are mandatory to band-limit the spectrum of signal to the prescribed bandwidth in harmony with the actual sampling rate. In sampling rate conversion systems, the filters are used in decimation to quell aliasing, and in interpolation to refrain from imaging. Since, it is not possible to attain an ideal frequency response, therefore the filter characteristics determine the performance of the system for sampling rate conversion. Thus, an appropriate and cost effective filter design is required for the sampling rate conversion with the minimum signal distortion. Understanding sampling and analog reconstruction is very pivotal in order to realize the multi-rate systems. In fact, the sampling of analog signal is just the extreme case of down-sampling a digital signal, thereby reducing its sampling rate. Similarly, the analog reformation is only the edge case of raising the sampling rate of a signal.

If the required sample-rate is an integer multiple of the existing one, it is adequate to oversample the input signal with the help of an interpolation filter. Similarly, if the existing sample-rate is an integer multiple of the one that is desired, then the process of decimation or down-sampling can be commissioned. However, if the ratio of input and output sample-rate is an arbitrary number, matter becomes more confounded. In digital audio players, the arbitrary sample rate conversion is becoming more and more general, although the deliberations for doing this are rather scattered. One argument often used is that the up-sampling improves the sound quality by easing the anti-alias filter requirements. Considering the fact that most of the digital to analog converters (DACs) already have at least 64 times oversampling, the significance of this is very much doubtful. The commercials often depict it as if up-sampling magically improves the signal quality, while in actuality, it introduces losses. Arbitrary sample rate conversion

unavoidably adds distortion, although in very small amounts even if the sample rate converter is well-designed. Thus, there is a need of efficient filter design for the arbitrary sampling rate conversion with minimal signal distortion.

In case of higher rate change, the multi-stage implementation is preferred over single-stage implementation for most of the filtering architectures. But, the multi-stage implementation enhances the computational as well as implementation complexity of the overall communication system. Also, there are a large number of parameters and trade-offs involved in the design of multi-stage interpolation systems, including the choice of the number of stages, the interpolation ratios at each stage, the filter requirements and the actual filter designs at each stage. Therefore, it is imperative to find out some techniques by which the complexity of multi-stage implementation of the filtering structures can be curtailed.

1.2 MOTIVATION FOR THESIS

Wireless broadband data communication is experiencing a swift development. To cope up with this remarkable growth in the requirement for wireless broadband, new technologies and architectures are required to improve the system performance and network scalability while considerably tumbling the outlay of equipment. With increasing cost pressures on the wireless equipment makers, there is a significant impetus to limit both capital expenditure and operating expenditure attributes of the communication systems. The basic building blocks in the modern digital communication systems are entities those either increase or decrease the sampling rate called interpolators and decimators respectively.

In this thesis, the primary aim is to design and implement efficient structures for the digital interpolators. To accomplish this goal, the cost efficiency and optimal performance designing for the interpolation and decimation filters are of main concern. Therefore, the

detailed literature survey is carried out to investigate the design techniques, which can lead to the development of efficient and low cost interpolation filters.

Almost all filter banks execute the operations of up-sampling, linear filtering, down-sampling, and therefore belong to a class of linear, periodically time-varying systems [3]. However, it is possible to change the ordering of these operations, due to its uniform, time-varying property [4]. The multi-rate filter design, under various aliasing and cross-talk free constraints, has been the main focus of study in the filter bank theory, and it is mostly modeled through the matrix notations. As compared to the individual single channel filters, the filter banks are efficient signal processing structures both in terms of the design and computation [5]. Despite all this, the implementation cost may be a consequential factor in the realization of modern communication systems.

1.2.1 Sample Rate Conversion (SRC) using Cascade Integrator Comb (CIC) Filters

Hogenauer [6] proposed a class of digital linear phase finite impulse response (FIR) filters for the decimation and interpolation, which require no multipliers (use limited storage), making them a viable substitute in the conventional implementation for certain applications. The CIC architecture can be used for both interpolation and decimation. In the CIC filter design, the pass-band response and stop-band rejection characteristics are managed by three-integer parameters, which are number of stages, the differential delay and the number of bits in input/output registers. For example, higher the number of stages, higher will be the existing stop-band attenuation.

The CIC filters are multiplication free filters with limited storage requirements, which make them ideal for the high speed data converters. Hogenauer [6] presented an FIR structure, which consists of cascaded integrator stages working at the higher sampling rate; and the same number of comb stages working at the low sampling rate. A number of cascaded integrator comb pairs are chosen to meet the design requirements for aliasing or

imaging errors. Although, the CIC filters can implement decimation and interpolation efficiently in the hardware for a wide range of rate change factors, yet CIC filter response is lacking in a flat pass-band response and better transition bandwidth. To circumvent these problems, a compensation FIR filter can be employed in cascade with the CIC filter to provide frequency correction as well as spectrum shaping.

Perceptible efforts have been made to construct the multi-rate solution based on the CIC filters [7]. It is more conducive to use the multi-stage CIC filters for the large decimation ratios. To offer superior alias rejection, a modified CIC filter design has been proposed by Saud and Stuber [8]. The conventional CIC filters exhibit high pass-band droop. A separate FIR filter is required to compensate this droop. Jiang and Willson proposed a filter sharpening technique in [9] to improve the pass-band characteristic of CIC filters by using the multiple copies of the identical filter. Once again, this filter sharpening comes at the cost of excessive computations per second for the higher order CIC filters. A similar approach to the CIC filter sharpening has been proposed with an efficient structure by Dolecek and Mitra in [10]. For a factorizable integer decimation factor, a two stage realization has been propounded with filter sharpening using the multiple copies performed in the second stage at low sampling rate, while the first stage of comb filters is poly-phase decayed to increase effectiveness. It was presumed that the CIC filter response can be decomposed into the separate comb and integrator sections for a specific decimation ratio. But, this proposition may not hold in practice for the multi-standard digital communication systems, which support various wireless standards like WCDMA, WiMax, where diverse symbol rates must be regulated.

The architecture for SRC unit has been designed and implemented by Savvopoulos and Antonakopoulos in [11], which can be integrated into the next generation software radio receivers. The presented multi-rate module is based on the two cascaded stages of

frequency conversion, which operates on the samples fed by an A/D converter of moderate sampling frequency. The architecture also contains the SRC unit, which performs the operation of frequency conversion on the resulting in-phase and quadrature (I/Q) samples for the proper interfacing with demodulator at the low rates. Babic et al. proposed a proficient way in [12] to realize flexible multi-rate signal processing systems with high oversampling ratio and sampling rate conversion ratio. The proposed multi-rate filter comprises of parallel CIC filters followed by a linear interpolation filter. The principle is to employ the two parallel CIC filters to analyze the two desirable sample values for the linear interpolation. These samples arise just before and after the last output sample, but this corresponds to a system, where the linear interpolation is implemented at the high input sampling rate. Due to this much high rate, the execution complexity may augment. Although, it can be used in a multi-standard communication receiver, yet sometimes there may be a problem of symbol rate synchronization. Chapman has revealed in [13] that the implementation of digital filters with sample rates, just a few MHz above, are generally complicated and exorbitant to realize using the standard discrete-time signal processing.

Hentschel and Fettweis proposed a multi-standard SRC solution in [14] for the CIC structures, where the SRC is performed after initial decimation using FIR filters. It is meant to be a reference solution exploring the advantages and issues of the different alternatives. However, the CIC filters are characterized in nearly all promising solutions of the proposed multi-standard SRC scheme, in one form or the other. The decimation before CIC filter structure reduces the sampling rate, so that the polynomial filters computations are performed at the low sampling rate. However, it is not always possible to shift polynomial interpolation to the lowest sampling rate as the arbitrary decimation preceding to it, which can result in the aliasing errors. A modified form of this structure

was proposed by Saud and Stuber in [15] to offer better alias rejection by spreading the delays in CIC comb stages, which in turn increases the stop-band attenuation. The results demonstrate that the modified CIC filters outperform the conventional CIC filters for the purpose of SRC in the software radio systems. It was shown that this amelioration in performance comes at little computational and memory overhead for the fractional sample ratios near unity. Kaiser and Hamming proposed a sharpening technique in [16] to improve the pass-band characteristic of non-recursive filters by means of the multiple copies of the same filter. However, this filter sharpening comes at the cost of excessive computations per second, especially if the order of basic non-recursive filters is high. Using the recommended technique, the need of a programmable decimation filter arises, following the non-recursive stages those help in the designing of a single chip digital programmable decimation filter.

Another low complexity approach for the CIC compensation is through low order frequency-domain polynomials to improve the pass-band response. This approximation has been described in detail by Oh et al. in [17]. A minor defect in this scheme is significant increase in the computational cost, if a polynomial of order more than four is required to correct the pass-band droop of CIC filter. Other than this, the proposed scheme is fairly striking, particularly for the compensation purposes. The poly-phase structures for CIC multi-rate filters are proposed by Yang and Snelgrove in [18]. With such novel structures, the projected filters can work at much less sampling rate, yet attain approximately the similar performance as Hogenauer's [6] CIC filters. It demonstrates the advantages of proposed filters under the high speed operation and low power consumption.

A power efficient multi-rate multi-stage filter for the mono-bit and multi-bit sigma delta analog to digital converter (ADC) is also presented in [19]. The poly-phase

decomposition in every stage, with very high SRC factor in the primary stage, is used to meaningfully reduce the sampling frequency of comb filter. Numerous implementations specify that the appropriate choice of the first stage SRC factor can significantly improve the power consumption, area and the maximum sampling frequency. The linear and non-linear distortion methods inside the digital audio FIR low-pass filters used for the decimation and interpolation have been explained by Dunn in [20]. In this manuscript [20], four filters have been contrived using the Remez algorithm. The typical pass-band ripple attribute has been analyzed to approximate the time-dispersion properties of the filter within the audio band. This investigation is used to explain the considerable dissimilarities in the pre-echo and post-echo performance among the illustrated filters. The vulnerability of filters and the effect of conciliations in stop-band performance close to the folding frequency are deliberated.

The filters intended for the high sampling frequency are used to demonstrate that how all these distortion methods can be abridged by filters of comparable computational requirements, but with a more soporific transition area. This results in either less ripple (and hence pre-echo) amplitude or in a lower time displacement for the echo. In both the cases, the result is probable decrease in the audibility of the echo. A straight consequence of the high sampling rate is that the time displacement will scale inversely proportional to the sample rate for an identical filter design. Hence, the advancement can be made just by increasing the sample rate. More effort is necessary to evaluate the limits on the awareness of echo effects described by Dunn in [20]. It should envelop both the audibility of echoes and their consequences on the localization of sound origins. The effects described in [20] point out that it may be complicated to differentiate any valuable effects of an enhancement in the sampling frequency from the dissimilar filter performance; this must be kept in mind, while making comparisons among the different sampling rates.

An arrangement for the implementation of N^{th} order 1-D all-pass digital filters having N multipliers has been presented by Reddy et al. in [21]. The N^{th} order all-pass transfer function is understood as whole as opposite to the implementation using cascade of the first or second order filters. But, there was previously a technique proposed by Mitra and Hirano for realizing N^{th} order all-pass filters. Though, this methodology can be used to realize the N^{th} order transfer function as a whole, it is rather wearisome to do so. Reddy et al. in [21] proposed a framework for realizing N^{th} order 1-D all-pass digital filters by following the approach used by Manivannan and Eswaran in [23] to realize the 2-D all-pass filters. This method is straightforward, and the consequential structure is having N multipliers and N unit delay elements.

A novel architecture for the realization of high-order decimation filters has been proposed by Kwentus et al. in [24]. It combines the CIC multi-rate filter structure with filter sharpening techniques for improving the filter's pass-band response. This permits the first-stage CIC decimation filter to be pursued by a fixed-coefficient second-stage filter, therefore it attains a significant hardware reduction over prevalent ideologies. Additionally, the application of fixed-coefficient filters in place of the programmable coefficient filters improves the total throughput rate. The resulting architecture is well-matched for the single-chip VLSI realization with the very high data-sample rates. It is very advantageous for such applications which include all-digital sub-band tuning for the wideband communication links and signal analysis. Moreover, this enhancement in the performance comes at the cost of more computational and memory overheads.

1.2.2 Arbitrary Sample Rate Conversion using Farrow Structures

Farrow proposed a technique using FIR filters in [25], which amalgamates a controllable delay. Echo cancellation, symbol synchronization, etc. are the various applications of this suggested structure. Also, the proposed interpolator filter structure can also be used to

reconstruct the original band-limited signal from the samples taken at the Nyquist rate. The interpolation filters were proposed by Vesma in [26], which make use of the modified Farrow structure. This structure is an alteration of the original Farrow filter structure in such a way that this configuration of filter is a linear combination of the linear phase FIR sub-filters. When it is fixed, the overall filter approximates an all-pass filter with the fractional delay, provided that the sub-filters have been designed in a precise way. The main advantage of using the Farrow filtering structure is that the sub-filters do not have to be redesigned, when a new fractional delay is required. The filters with adjustable fractional delays are a necessity in the umpteen applications. One example is SRC using the arbitrary conversion factors, where the conventional interpolators and decimators used for the integral rational-factor conversion are unsuccessful or involve very high interpolation and decimation factors.

Ramstad put forth filtering methods in [27] for interfacing the discrete time systems with varied sampling frequencies. These methods are appropriate for the sampling rate conversion among any two sampling frequencies. The proposed practical implementations are either based on analog reconstruction filters, where the resulting digital filter coefficients are functions of the distances among current input and output samples or the digital interpolators combined with the simple analog interpolation schemes for finding the desired value in between the uniform output samples from the digital interpolators.

Cucchi et al. [28] presented a revision of Ramstad hybrid structure [27] for the arbitrary sampling rate conversion, which uses the B-spline function as analog re-sampler. The proposed technique proved to be quite effective in terms of the real time conversion among the standard frequencies of the digital audio using a single chip explication. Tseng proposed a new variable fractional delay filter design method using the

differentiator bank in [29]. In this, the Taylor series expansion was used to transform the specifications of the variable fractional delay filter into one of the differentiator banks. Thereafter, a window method is projected to design the differentiator bank.

Digital fractional delay (FD) filters find their application in the multitudinous applications [30]-[33]. In some of the cases, it is preferred that the FD must be adjustable, i.e., it can be altered during operation without re-designing the filters [34]. Alternatively, one can make use of the Taylor series of the desired (ideal) frequency response. From the Taylor series, it is evident that the required frequency response can be approximated by the modified Farrow filter structure. A variable fractional delay (VFD) filter is extensively used in applications such as symbol timing recovery, the arbitrary sampling rate conversion and echo cancellation. Nithirochananont presented an implementation of VFD filter on the field programmable gate array (FPGA) in [35]. This implementation utilized an efficient structure, called Taylor structure. The principal advantage of this structure is to reduce the number of multipliers and adders, as compared to the Farrow structure or the modified Farrow structure.

An unconventional efficient method for the conversion between arbitrary sampling rates has been suggested by Babic and Renfors in [36]. This method allows arbitrary number of zeroes at the multiples of both input and output sampling rates. This implementation has been proposed for the modified Farrow as well as transposed modified Farrow structures. A closed form expression for the continuous frequency response of the Lagrange interpolators is presented in [37]. The continuous frequency response is one of the most useful characterizations of an interpolation system. The projected expression holds true for the arbitrary interpolation orders and allows efficient as well as numerically stable evaluation. Until now, several properties of the Lagrange interpolators such as zero locations are described only pragmatically. Due to the brief

form projected formula, these characteristics can now be represented in an exact way. But, there is a need to apply the proposed closed form results to the wide scope of structural characterizations of the digital interpolators.

Some important and significant system options developed by Dick and Harris in [38], [39] are at disposal of the designers as spin-offs of the derivation of Farrow filter, in which the authors have derived the Farrow filter structures, which support the variable re-sampling continuously. A fractional delay filter is a tool for performing the band-limited interpolation among the samples of a time series. The Farrow filter is a multi-rate filter structure that offers the choice of continuously adjustable delay. However, this will be helpful only for the applications having fractional sample rate conversion [38], [39]. Valimaki proposed a new filter implementation strategy in [40] for the Lagrange interpolation for a maximally flat FIR fractional delay filter, in which the author considered the well-established filter design theories and suggested new dimensions to search for the efficient solutions for the arbitrary sample rate conversion.

Cook et al. generalized the well-known variable fractional delay filter structure in [41] for the Farrow filter by using Laguerre and Kautz filtering concepts. Here, the same authors also showed that how one can constitute amplitude response determining the optimization design formulation; and also protracting the range of variable delay. A few prominent features of the new structures are established through the three design examples based on the simple least squares criterion. Methods to design the fixed filters have been encapsulated in [25], and some extended research work has been reported in [42]. Ngia has also demonstrated in [43] that how the conventional Farrow structure can be generalized to encompass Laguerre and Kautz filtering concepts. The Laguerre and Kautz filters are high order forms of the unit delay elements of an FIR filter. By substituting the unit delays in Laguerre and Kautz filter sections, an extra one or two

degrees of freedom is incorporated in the design. A technique for generating the Farrow filter coefficients that minimizes the mean square error (MSE) at the symbol decision instants has been presented by Watkins in [44]. Minimizing the MSE at the symbol decision instants is almost equivalent to minimizing the symbol error rate (SER), but it is mathematically more expedient. The MSE, at time instants, other than the symbol decision instants does not influence the SER, and it is also not measured in the optimization.

In signal processing applications, it is quite often essential to attain the higher-order derivatives of data, especially at low-frequency range [45]. So far, numerous techniques have been developed to design FIR and infinite impulse response (IIR) digital differentiators, such as the Remez exchange algorithm [46], the eigen filter method [47], the maximally flat method [48], the weighted least square method [49], the quadratic programming method [50] and the Taylor series method [51] etc. Recently, the designs of two-dimensional (2-D) and fractional-order FIR differentiators have also acknowledged huge attention [52]. On the other hand, the fractional sample delay has turned out to be a significant scheme in the field of time adjustment in digital receivers, antenna array processing, speech coding and synthesis, modeling of music instruments, A/D conversion, etc. Till now, numerous techniques have been developed to design fixed fractional delay (FFD) and VFD filters. In the case of FFD, the delay value is fixed, therefore traditional FIR and IIR filter design techniques can be employed to devise FFD filter directly. In case of VFD, there are two types of design methods. One is the VFD FIR filter design and the other is the VFD all-pass filter design. The approaches developed to design VFD filter contain the Farrow method, the weighted least square method [53]-[55], the constrained mini-max optimization, the eigen filter method [56] and the singular value decomposition (SVD) method [57].

Tseng presented the limit computation method in [58] to design the digital differentiator by using the fractional delay filter. First, the relation among fractional delay filter and the first order differentiator is incepted, such that the differentiator can be designed with the help of fractional delay filter. Thereafter, the traditional FIR, all-pass and Farrow based fractional delay filters are directly applied to devise the first-order differentiator. Next, the proposed method is well extended to design high-order differentiators. Finally, numerous design examples are demonstrated to show the effectiveness of this design approach. Satish and Gupta [59] proposed a new method for the design of computationally efficient non sub-sampled multi-scale multidirectional filter bank with perfect reconstruction (PR). The potential application of the projected filter bank is demonstrated by juxtaposing the image de-noising performance of the proposed filter with other design method that exists in the research archives.

Harris [60] presented a derivation of the technique proposed by Farrow, and also presented the performance and complexity of re-sampling filters using this technique. The work also evolves a few vital system options available to the designer as spin-offs of the derivation. A technique to interpolate a bounded band-limited signal from its own samples, with minimum complexity, has been proposed by Selva in [61]. The technique was a result of combining a rapidly converging sampling expansion with the Farrow interpolator technique. For the Farrow interpolator, it assures that the interpolation error is smaller than a known bound, both in time as well as frequency-domain. Also, the method is enlisted to give a simple solution to an important efficiency conundrum in the simulation of communication system and applications [62], [63].

Lethian et al. [64] discussed and equated the properties of continuous as well as discrete-time models in optimization of filters for the sample rate conversion, so that the appropriate model can be chosen for each filter design task. The optimized interpolator

using the discrete-time model is the best, when a rational conversion factor is fixed. However, the optimized interpolator using the continuous-time model can be more robust against changes of the conversion factor, the property which is useful in applications that require flexibility. A diversity of approaches to the design of generalized interpolators have been described in [65], [66]. However, the interpolators proposed in these research archives are mainly based on the classical design methods, which are not always befitting to the generalized interpolation problem. Also, the investigation methods in [65], [66] focus on the time-domain measures of the interpolation error, rather than the types of frequency-domain considerations, which are vital for some applications. Adam et al. discussed some unconventional optimality criteria in [67] for the generalized interpolators. These optimality criteria are particularly significant for the polar format signal processing in the synthetic aperture radar systems, in which the same author also demarcated and justified the frequency-domain optimality criteria for the generalized interpolators and described an approach for designing the corresponding digital filters.

Goslin has elaborated the importance of the FPGAs in [68] for application specific uses. The FPGAs have turned out to be a competitive substitute for the high performance digital signal processing (DSP) applications, previously preponderated by the general purpose DSP and application specific integrated circuits (ASIC) devices [69], [70]. Dick and Harris have investigated an architectural option in [71] for constructing an interpolation filter with the integer factor using Xilinx FPGA technology. Although the proposed architecture helps in saving the device logic resources as compared to the other methods that offer the identical functionality, yet this architecture is limited to the narrow band filters only. Lokken proposed SRC with separate source and sink clock signals in [72]. Although, such SRC can offer very good jitter rejection, but still it subsumes some distortion and noise to the signal.

1.2.3 Efficient Implementation of Polynomial Based Interpolation Filter

The polynomial based filters are time-varying filters, whose impulse response is a polynomial in the time-domain. These filters are ideally appropriate for the digital interpolation as described by Erup et al. in [73], especially when implemented using the Farrow filtering structures. The time-varying nature of polynomial based filters require pre-computing and registering the impulse response. The Farrow filtering structure helps to evade this by allowing the impulse response to be calculated online. In software defined radio (SDR) systems, the polynomial based filters are ruminated to be the architecture of choice for the fractional SRC, when compared with the traditional implementations, where prohibitively high intermediate sampling rates are common, all the computations are undertaken at the input sampling rate in Farrow filtering structures.

The Farrow implementation of polynomial based filter has better anti-imaging attribute, and it is therefore suitable for the interpolation. However, a modified structure called the transposed Farrow structure has also been proposed that exhibits the anti-imaging hallmarks suitable for the decimation. Though the transposed Farrow structure requires only a slight increase in the computational complexity (an extra integrator stage is required) as compared to the original Farrow structure, but the multiplications measured directly at the ADC sampling rate are relatively high when compared to the CIC filters which do not require multipliers. Ghadam et al. proposed a polynomial approximation for CIC filters in [74]. It has been suggested that both Farrow and transposed Farrow structures can be used for implementing the polynomial approximation of CIC filters. The resulting structure needs less multiplications, and also has both good anti-aliasing and anti-imaging behaviour. A multi-standard SRC solution was presented, in which a Farrow structure performed fractional SRC after the initial decimation using the FIR filters. However, the decimation before Farrow structure reduces the sampling

rate, so that the polynomial filters computations are carried out at the low sampling rate. Moreover, it is not always possible to shift the polynomial interpolation to the lowest sampling rate, since arbitrary decimation prior to it may cause the aliasing enigma.

A comprehensive design technique for the polynomial based interpolation filters has been proposed by Vesma and Saramaki in [75]. These filters can be realized with the help of a modified Farrow structure, where the fixed FIR sub-filters have either symmetrical or anti-symmetrical impulse responses. In the projected technique, the piece-wise polynomial impulse response of the interpolation filter is optimized directly in the frequency-domain with the help of least mean square criterion subject to the given time-domain restrictions. The extent of impulse response and the degree of approximating polynomial in the polynomial intervals can be randomly chosen. The optimization in the frequency-domain makes the proposed design scheme more viable for the different digital signal processing applications, and it allows the user to create interpolation filters for the arbitrary desired weighting functions. Most significantly, the interpolation filters can be optimized in a way akin to that of the traditional linear phase FIR filters.

A filtering arrangement with linear complexity is projected for the Lagrange interpolation by Candan in [76]. This structure is analogous to the Farrow structure in principle, but it is more efficient and has the enhanced feature of being order updatable on-the-fly. The main purpose of the proposed structure is the realization of fractional delay filters to alleviate the symbol synchronization errors in the digital communication systems. Various other applications are time-delay evaluation, echo cancellation, acoustic modeling and the arbitrary sampling rate conversion. In digital communication systems, the receiver clock may have time-varying offset causing inter-symbol interference (ISI) at the channel output. One method for compensating the error is to fractionally delay the signal as suggested by Samadi et al. in [77]. The hardware optimization of Farrow

filtering structure and numerous other applications on the multi-channel signal sampling and reconstruction have been confabulated in [78], [79]. The proposed structure exhibits linearly growing computational complexity, whereas the previous proposals have complexities growing with the square of interpolation order.

A genre of multi-rate filters namely polynomial interpolation filters has also been used for an efficient implementation with digital techniques to correct the symbol timings, and also the effects of band-limited channels on the communication receivers. The band-limited channels results in ISI. This ISI can be mitigated by transmitting the symbols, which are pulse shaped by the root raised cosine filter that limits the frequency contents of source symbols. Before pulse shaping stage, the transmitted symbols are oversampled. A number of multi-rate realizations help to combine the pulse shaping and interpolation at a fraction of the cost as compared to the complexity of traditional interpolation operations [80], [81]. Yoon et al. [82] presented a hybrid structure of the transposed direct form (TDF) and the direct form (DF) filter based on the flattened coefficients technique, so that it can reduce the number of flip-flops and full-adders without the additional critical path delay. The "resource sharing method" and the "sharing pattern searching" algorithm have also been presented to decrease the number of adders without altering the logic depth.

The polynomial based FIR filters with adjustable fractional delay were proposed by Vesma and Saramaki in [83]. The major advantage of the projected synthesis scheme lies in the fact that it results in the structure, where the linear phase branch filters of the modified Farrow structure are fixed, and only one parameter controls the fractional delay. An efficient approach for optimizing the polynomial based FIR filters with adjustable fractional delay has been introduced by Vesma and Saramaki in [84]. Here, these edifices can be efficiently implemented using the modified Farrow structures. Vaidyanathan [85]

presented some of the less known aspects of sampling, which gave prominence to the point wise stability of the reconstruction, and also the reconstruction from the non-uniform samples, which finds its applications in the multi-resolution computation and in the field of digital interpolation.

A new mode for the design of bi-orthogonal filter bank with improved frequency response was proposed by Patil et al. in [86]. The filter bank was designed using the modified polynomial factorization approach. It has been shown that the filters with better degree of freedom can be designed by increasing the order of remainder polynomial. The results show that the frequency response of the filter can be improved using the higher order remainder polynomials keeping the similar regularity. Babic introduced an approximation using Taylor series in [87], that is implemented as piecewise polynomial function, where in each interval f(x) is approximated by a Taylor polynomial of the low order. The piecewise polynomial approximation can be efficiently realized using the Farrow filtering structure.

For flexible designs, it is much better to rely on the all-digital solutions. The Farrow filtering structure has earlier been successfully applied to the problem at hand, and a modified Farrow structure including the poly-phase filters was evolved in [88]. The same structure is also known as the poly-phase Farrow structure (PFS). Ramstad propounded a method for changing the sampling points of an analog low-pass signal in [89]. These are based on polynomial approximation of subintervals of the impulse response of an analog reconstruction filter. The PFS is used for the efficient implementation of the filter. The method portrays great flexibility to adapt and optimize for the different applications and hardware. The poly-phase filter bank based, digital up-converter and digital down-converter have been presented by Kiessling and Mujtaba in [90]. In this research work, a generalized poly-phase filter bank incorporating the sample rate alterations has been used

as a wideband digital up-converter at the transmitter, while the corresponding filter bank does the transpose operations at the receiver. However, the flexibility of the proposed architecture is restricted to a distinct standard, since the sampling frequency is mainly dependent on the channel spacing and standard's baseband symbol rate. Although, it annihilates the requirement of separate sample rate converters after the channelization, yet this method increases the rate of computations in the filter bank in order to achieve the arbitrary sample rate conversion.

The strict operational conditions are imposed by the poly-phase discrete Fourier transform (DFT) filter bank, such that it is usable with only one standard at a time, which motivated Saud and Stuber [15] to incorporate the complex modulated PR filter banks. In their proposed scheme, the wideband input signal containing the variable bandwidth channels has been decomposed into the sub-channels. The sub-channels have been grouped into mutually exclusive sets with a particular channel. This channel is extracted by reconstructing the corresponding sub-band sets through the synthesis filter bank by using an appropriate coding scheme. The drawback of this scheme is the information required by different synthesis sections to determine the sub-band components those combine to form a user channel. To an extent, it also hampers the flexibility of the design, since it furnishes variable bandwidths; but the user channel position within the sub-band is still fixed. Moreover, when compared to the uniform DFT filter bank, the signal-to-noise ratio (SNR) performance of PR channelizer may be better with the same number of taps in the prototype filter. Although a poly-phase form of the PR filter bank channelizer has also been derived, but it is applicable only to the number of channels greater than fifteen. The simulation results demonstrate that the simple interpolator provides extraordinary performance. In many cases, the two-point linear interpolation is adequate. If better performance is desired, the classical four points and third-order polynomials can

be utilized. Moreover, a novel four-point interpolating filter with the piecewise-parabolic impulse response can have performance better than that of the standard interpolator, and still be implemented in a simpler way.

A narrow-band sigma-delta audio DAC having a digital signal interpolation part with 5-bit sigma-delta modulator in the final stage was proposed by Ameur and Loulou in [91]. The applications of this type of DAC are in the area of high quality sound applications, sigma-delta data converters and maximum likelihood (ML) equalizers [92]. To decrease the sensitivity of DAC non-linearities of the last stage, dynamic element matching (DEM) was introduced. Ameur and Loulou [91] presented a MATLAB description modeling approach for the proposed DAC architecture. The sigma-delta modulator design can be configured as the third or fifth order, and it allows 24-bit pulse code modulation (PCM) at the sampling rate of 64 kHz for the digital versatile disk (DVD) audio application.

A few techniques have been covered in literature on the direct conversion [93]. The direct conversion does not involve the digital intermediate frequency (IF) stage. The aliasing is sometimes referred to as the under-sampling, which has been explained by Seo et al. in [94]. It was depicted that if IF and the sampling frequency of a band-limited user channel are selected suitably, it is possible to down-convert this channel without complex multiplications, provided the sampling frequency is at least twice of the user channel bandwidth. As switching among the different clocks can introduce jitters in the signal, therefore a single clock is desirable in any digital communication system [95]. Moreover, a single clock system has the lower power consumption as compared to the digital communication system with multiple clocks. Dengwei et al. developed an innovative interpolation technique in [96], which instead of approximating the continuous-time signal with any traditional polynomial and computing the synchronized samples with the

help of a Farrow structure, utilizes the trigonometric polynomial. The simulation results designate that the enhanced performance reduces the computational delay, and in most of the cases, streamlined hardware can be attained. This method was formulated in terms of a digital interpolation filter. It has been observed that with a minor modification of the original technique, the interpolation filter based on the trigonometric polynomial can achieve a required frequency response. Francesconi et al. suggested a novel approach in [97] to implement the interpolation function by means of Lagrange interpolators. The proposed architecture does not require the high frequency multipliers. This structure improves system performance at the cost of a small enhancement in the circuit complexity.

Choi and Munson derived a closed form for the interpolation error in [98]. This expansion shows that the error depends on the eigen value distribution of the matrix, which is specified by the set of sampling points. The suggested interpolator is easier to implement. The impulse response of the polynomial based interpolation filter has been responded as the continuous-time, piece-wise polynomial function known as the Farrow filtering structure, which has assorted modifications. The polynomial based interpolator filters have been optimized using hybrid analog/digital models. However in practice, these filters are implemented in digital systems, and they have corresponding FIR equivalent filters. In [99], a discrete time model was derived for the zeroth-order polynomial interpolation. Further in [100], it is reported that for any modification of the Farrow structure, there exists a discrete-time FIR filter. Babic proposed methods for the discrete-time optimization of polynomial based interpolation filters in [101], and the author also compared properties of the filters optimized in traditional way using the continuous-time model, and the filter is optimized using the discrete-time model. These

methods are based on relation between the polynomial based interpolation filter and the corresponding FIR filter paradigm.

The best synchronization algorithms for the digital communication systems are based on the ML estimation theory. Evaluation of the carrier phase and the symbol timing can be achieved separately or jointly. However, the estimates obtained from joint optimization of the likelihood function are typically superior [102]. Recently, a novel method for clock and carrier phase synchronization of the digital receivers, using block based feed-forward architecture, has been introduced in [103], [104]. This technique employs a practical and digitally developed synchronization model, using interpolation technique, for jointly evaluating the symbol timing and the carrier phase. Here, the ML function is expressed in terms of the polynomial-based interpolator filter branches. Hamila et al. analyzed the effect of frequency offset or Doppler shift on the performance of the digital receiver in [105], while adjusting the symbol timing and carrier phase. Here, the frequency offset is much lesser than the symbol period. Though it eliminates the need for separate sample rate converters after channelization, but this method increases the rate of computations in the filter bank in order to achieve the arbitrary sample rate conversion. Therefore, there is a requirement for more generic approach towards the structural manipulation of the digital interpolators.

1.2.4 Cost Effective and Efficient Realization of Multi-stage Interpolation Filter

Aware of the fact that a multi-stage design is more innovative than its single-stage counterpart with respect to the computational estimation and memory requirements, the purpose is to determine the optimal number of stages for a particular sample rate change factor. The determination of a number of stages in the multi-rate filter design has been formulated and solved as an optimization problem in [106]. The optimal multi-stage design with respect to computations also leads to a design with the least memory storage

necessities [107]. This is due to the fact that the storage requirements mostly depend on the filter order, which strongly affects the number of computations per second. Therefore, minimizing computations also optimizes the design with respect to the storage requirements. Although, this approach provides less memory requirement, yet it does not yield a method by which the filter can be implemented with a few resources on FPGA platform. The coefficients of a half-band filter can easily be obtained using the traditional algorithms [108]. The half-band filters appear in conjunction with CIC and linear phase FIR filters to complete the sample rate converter design [109]. But, the half-band filter implementations are more useful in the multi-stage applications, when the rate change can be factored by 2. Vaidyanathan and Nguyen has proposed filter design technique based on M^{th} band filters in [110]. Due to its simple design techniques, the half-band filters (M^{th} band filters with M=2) is a popular pick, in which the same authors also proposed that how the design time for the equi-ripple half-band filters can be reduced to a considerable amount. Extension of this method to M^{th} band design is also utilized. The new design method is considerably faster than the conventional approach for the half-band filters.

Chou et al. described an approach for the realization of digital filter algorithms based on FPGAs in [111]. The general purpose DSP implementations often lack the performance essential for the moderate sampling rates, while ASIC approaches are confined in flexibility and may not be cost effective for many applications too. The examples of FIR and IIR filter implementations demonstrate that the FPGA approach is flexible and offers performance comparable or superior to the conventional approaches. Because of the programmability of this technology, the examples presented by Chou et al. in [111] can be extended to offer a variety of other high performance FPGA based filter realizations.

Bellanger et al. derived in [112] that the redundancy in extrapolation and interpolation can be removed by introducing the half-band non recursive filtering elements, for which the definitions, the performance figures and the efficient implementation are supplied. The computational complexity and the storage requirements are reduced drastically, which bring the digital filters in the most favorable position for their competitiveness against analog filters in many application fields. Bellanger [113] presented estimations for the computation rate and the volume of storage needed in sequences of the half-band FIR filters used for the sample rate reduction or increment. It has been suggested that the accurate and quick evaluation of the half-band filter approach is helpful for the general design and can serve for comparison with other multi-rate techniques.

Goodman presented structures based on the half-band filters in [114], which may be used as components of the multi-stage interpolators. He also furnished a procedure for combining the filters to produce multi-stage designs that meet a very wide range of accuracy requirement. The same author also presented a simple re-sampling technique that extends the range of designs for conversion between any two rates. The interpolation ratio does not require being an integer or even rational term. In fact, it can vary slightly as in a practical situation, where both input and output signals are dependent on the autonomous clocks. Although an interpolator with many half-band filters is computationally efficient, yet it requires more elaborated timing and control circuitry than a scheme with fewer stages. The current inclination in the design of audio interpolation filters is focused on the utilization of FIR based approaches. The digital filters proposed in [115], for high quality audio DACs use a multi-stage cascade of comb and FIR filters. The low power audio delta sigma D/A converter proposed in [116], [117] uses a cascade of the half-band FIR filters in order to acquire the area optimized design for the interpolation filter. The modified merged delay transformation (MDT) based interpolator

architecture, appropriate for area efficient, low power digital audio applications has been presented by Mukhtar et al. in [118]. In addition to its hardware efficiency, it also has the advantage of reduced group delay in comparison to the existing FIR based approaches, which is desirable in many live audio applications.

Wang et al. [119] proposed designs of interpolators, including the two-time, four-time and eight-time data interpolators. In these designs, the same authors adopted the parallel architecture to realize the circuits for making the higher speed interpolators. To balance speed and accuracy of the interpolators, the eighth-order sinc function has been used as the interpolation function for utilizing the symmetry of sinc function. The chips of the proposed design are suitable for the high speed digital signal processing, image processing and other applications [120]-[122]. The new interpolator chips can ease the speed requirement of ADC and also facilitate the DSP for many high performance digital signal applications [123]-[126].

Crochiere and Rabiner presented arguments in the design of multi-stage decimators and interpolators in [127], in which the authors proposed that the designs, which are optimized in terms of the minimum storage requirement, are essentially optimized in terms of the computation as well. A realistic scheme for implementing the multi-stage decimators and interpolators was presented. Finally, the use of multi-band filter designs and IIR filter designs for decimators and interpolators was deliberated upon. Turek [128] demonstrated an interpolation system into numerous cascaded stages, the utilization of recursive as well as non-recursive interpolation filters, the use of linear phase and minimum phase interpolation filters. He also delivered a method to reduce the computational complexity with the increase in number of cascaded stages, but the proposed technique was not well-organized in terms of the efficiency.

1.3 PROBLEM FORMULATION

The modern digital communication system is experiencing a rapid expansion. To meet the stringent real time requirements posed by modern wireless communication systems, high performance and low cost signal processing architectures must be developed. Therefore, new technologies and architectures are needed to greatly improve system performance and network scalability while significantly reducing the cost of equipment. One of the functionality of basic building blocks of the modern digital communication systems is digital interpolation. The cardinal purpose of the interpolators is to increase the sampling rate. These systems are very helpful for the signal analysis, de-noising, compression and so forth. With increasing cost pressures on many wireless equipment makers, there is a significant drive to reduce computational complexity, power consumption, processing speed, flexibility etc. The interpolation can be efficiently realized with the help of cascade integrator comb (CIC) filters. However, the disadvantage of a CIC filter is that, its pass-band is not flat, which is undesirable in many applications. So, there is a need to introduce the techniques with the help of which the performance of interpolators can be improved at the different sampling rates.

However, the CIC filters are unsuitable for those practical applications, where the fractional delay is stringently required. As an interesting remedy for this problem, the Farrow filter structures are extensively used in practice for attaining the fractional delay. But, there is a scope to further reduce the computational complexity for obtaining appropriately reconstructed interpolated signal by using the Farrow filtering structures. The single-stage filters are efficient for the lower order interpolation factors. However, for the high rate change, the multi-stage filters are preferred because single-stage design does not offer effective response. However, the multi-stage design increases the complexity of the system. For such scenarios, there is a need to develop a technique for

the efficient multi-stage realizations of the interpolators. Moreover, better design and implementation of interpolators can result in the low cost communication system with improved performance.

1.4 RESEARCH OBJECTIVES

In the light of aforementioned aspects, the research objectives for the investigation are as follows

1. To explore the joint algorithm-architecture design paradigm available for performing sample rate conversion using digital interpolators.
2. To propose new architectures for digital interpolators to meet the real time signal processing requirements.
3. To functionally simulate and implement the proposed techniques for performance evaluation on FPGA platform.

1.5 THESIS OUTLINE

The thesis will be organized in six chapters. The details of the contents of each chapter of the thesis are as follows

- Chapter one presents the introduction and motivation based on the literature survey, problem formulation, objectives of the thesis and organization of thesis.
- Chapter two describes the effects of compensation and the arbitrary sampling in interpolators for the different wireless standards on FPGA platform. Also, it has been demonstrated that how the performance of multi-rate filters can be improved using different techniques at the different sampling rates. The filter structures suggested here have wide applications in the field of analog to digital converter (ADC), digital up-converter (DUC)/ digital down converter (DDC), and almost at every stage of the communication system, wherever the convolution operation is being performed.

- Chapter three delves into the designing of Farrow filter structure based interpolators using the Lagrange polynomial. The implementation has been performed using the cubic Lagrange and quadratic Lagrange polynomial, and the results have been compared. The proposed Farrow filter structure based interpolators have been designed using Altera DSP builder advanced block-set.
- Chapter four deals with the performance comparison of the quadrature amplitude modulation (QAM)/ quadrature phase shift keying (QPSK) based single-stage and double-stage digital interpolators, and parallels have also been drawn between them. The results present the efficient and effective implementation of the digital interpolation systems for the up-sampling of single-stage and double-stage digital interpolators. Contrast is performed in terms of the spectrum of generated signal, envelope power, transformed signal trajectory, noise performance, input and output constellation. The proposed interpolation filters have been simulated in the Agilent's advanced design system (ADS).
- Chapter five puts forth the efficient implementation of multi-stage digital interpolators using the half-band filters. For high rate changes, a multi-stage implementation of the interpolator is preferential. However, the multi-stage implementation increases complexity of the overall system. This complexity can be reduced considerably by using the half-band filters. All this has been proposed with the aid of Agilent's ADS.
- Finally, the chapter six sums up the research work. Further, a brief description about the future work/scope has been presented as a motivational seed for the germination of research work. The thesis concludes with the research publications as well as the list of references found useful during the course of investigation.

1.6 SUMMARY OF THE CHAPTER

This chapter acts as a capsule for the motivation of the thesis. An exhaustive literature survey has been presented about the sample rate conversion, arbitrary sample rate conversion, polynomial interpolation filter and multi-stage interpolation filters. Based on this, the objectives of the thesis have been identified, in which the main motive is to increase the efficiency and to reduce the computational complexity of the digital interpolators, and hence their cost. In the next chapter, the effects of compensation and arbitrary sampling in the interpolators for different wireless standards on the FPGA platform have been presented.

CHAPTER 2

EFFECTS OF COMPENSATION AND ARBITRARY SAMPLING IN INTERPOLATORS FOR DIFFERENT WIRELESS STANDARDS ON FPGA PLATFORM

2.1 INTRODUCTION

The digital filters have evolved as a strong option for removing the noise, for the shaping of spectrum and to minimize the inter-symbol interference in the communication architectures. These filters have become popular because their precise reproducibility allow design engineers to acquire performance levels those are hard to obtain with the analog filters. The main building blocks of any communication system are the multi-rate systems. Their purpose is to alter the rate of discrete-time signals, by adding or deleting a portion of the signal samples. A requisite component of the cost effective DSP algorithms is multi-rate signal type filters. Such filters utilize the extremely efficient structures to simultaneously perform the digital filtering, spectral translation, interpolation and decimation in both recursive and non-recursive structures. The FPGA has become an exceedingly cost-effective means to off-load computationally intensive digital signal processing algorithms for the progressive overall system performance.

The systems those employ multiple sampling rates in the processing of digital signals are called the multi-rate discrete time signal processing systems [129]. The cardinal purpose of the multi-rate systems is to either increase or decrease the sampling rate. These systems are very helpful for the signal analysis, de-noising, compression and so forth. These systems have increasingly created applications in the recent propensity in the areas of digital communication. The process of enhancing the sampling rate from the given rate by an integer factor (say '*L*') or an arbitrary factor is termed as the interpolation or sampling up-conversion. The interpolation [69], [113], [130] is the

process of up-sampling and filtering a signal to increase its effective sampling rate. The interpolators or interpolation filters are used at transmission side of the communication systems to increase the effective sampling rate of signal to be transmitted and to increase the data rate from baseband to the intermediate frequency (IF) signals. The figure 2.1 demonstrates the abstract view of transmitter, where the interpolation plays a vital role.

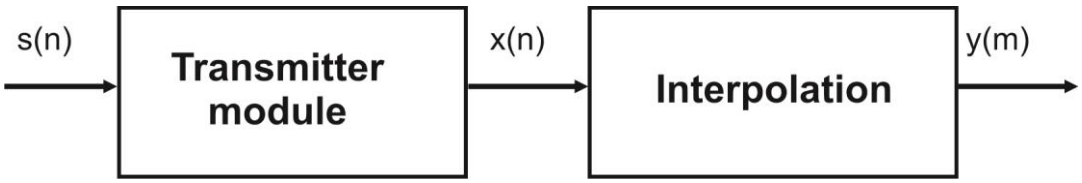

Figure 2.1: Abstract view of transmitter

where, *s(n)* is the input data signal at the transmitter, *x(n)* is the input to interpolator and *y(m)* is the output interpolated signal. Interpolation can be achieved by inserting *L-1* zero samples between the adjacent input signal samples [131]. The resulting signal has sampling rate $F' = LF$. While adding *L-1* new samples between each input sample increases the sample rate by a factor of *L*, and it also introduces images of the input spectrum into the interpolated output spectrum at the frequencies between the original Nyquist frequency and the higher interpolated Nyquist frequency.

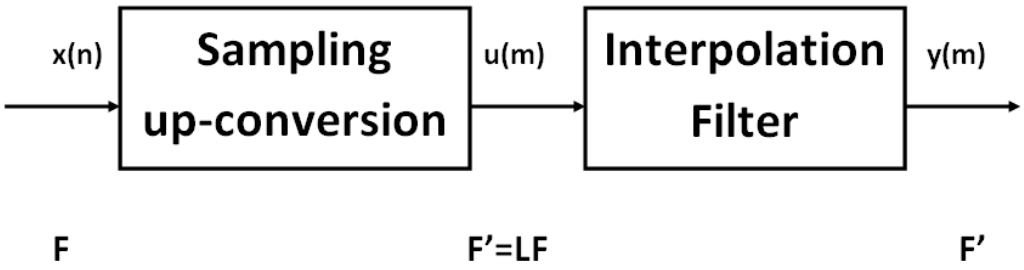

Figure 2.2: Block diagram of an interpolator

To mitigate the effect of imaging, the interpolated signal must be low-pass filtered to eradicate any image frequency, which will disrupt subsequent signal processing steps. A universal benefit of the interpolation process is that, the low-pass filter may be

implemented to operate at the input sampling rate rather than the faster output sampling rate. A block diagram of an interpolator has been shown in figure 2.2. If *x(n)* be the original input sequence and *u(m)* is the sequence with *L-1* zeroes inserted, then the output sequence *y(m)* of the interpolator is given by Rabiner in [132]. Mathematically interpolation can be expressed as

$$u(m) = \begin{cases} x\left(\frac{m}{L}\right), & m = 0, \pm L, \pm 2L, \dots \dots \dots \\ 0 & otherwise \end{cases} \quad (2.1)$$

when converting to the frequency-domain, *u(m)* is given as

$$U\left(e^{j\omega\prime}\right) = X(e^{j\omega\prime L}) \quad (2.2)$$

where, $U\left(e^{j\omega'}\right)$ is the Fourier transform of the signal *u(m)* expressed in terms of the spectrum of input signal *x(n)* and $\omega' = 2\pi F/F'$.

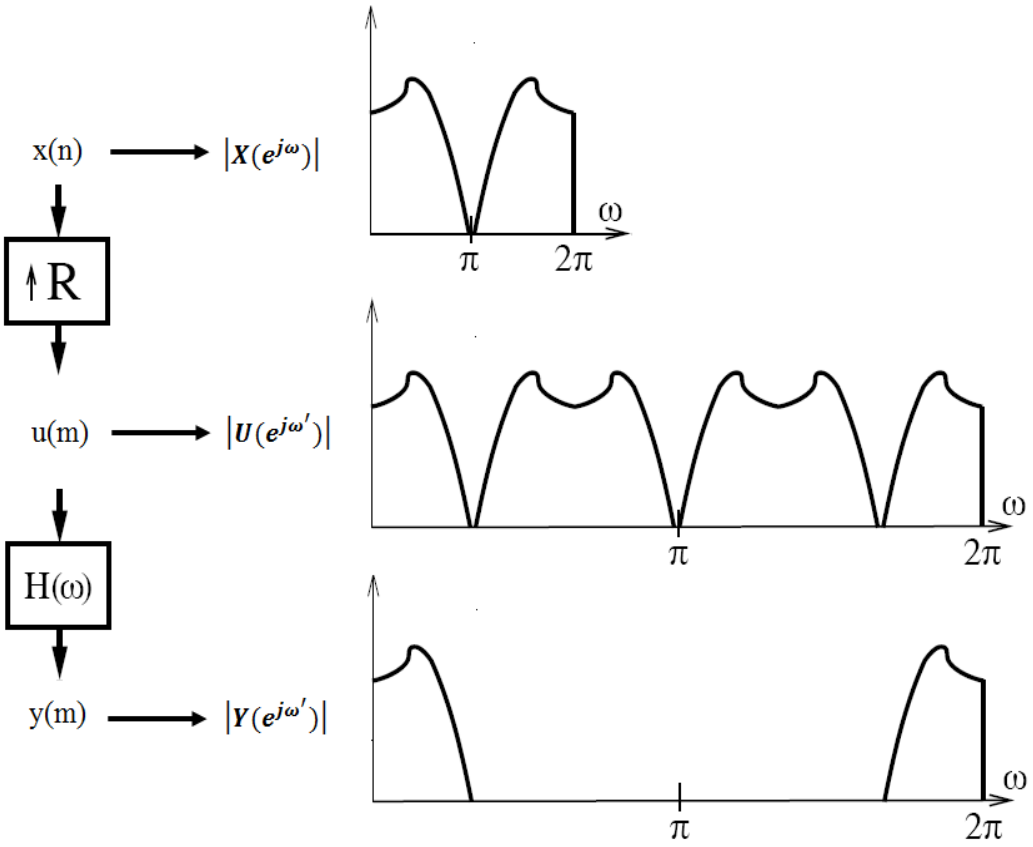

Figure 2.3: Typical waveforms and spectrum for interpolation by a factor *L=3*

The graphical representation of the interpolation process is depicted in figure 2.3. It consists of the input signal, the up-converted signal and the interpolated signal in time-domain and their respective spectrums. The spectrum of *u(m)* contains not only the required baseband signal frequencies (from $-\frac{\pi}{L}$ to $\frac{\pi}{L}$), but also images of the baseband signal placed at the harmonics of the original sampling frequency (at $\pm\frac{2\pi}{L}, \pm\frac{4\pi}{L}$, ...). To remove these unwanted harmonic components, it is pivotal to filter the up-converted signal *u(m)* with a digital low-pass filter, which approximates the characteristics of an ideal low-pass filter. The response of an ideal low-pass filter is given as

$$H\left(e^{j\omega\prime}\right) = \begin{cases} C, & \omega' \leq \pi/L \\ 0, & otherwise \end{cases} \tag{2.3}$$

Hence, the output interpolated signal in the frequency-domain for $C = L$ is

$$Y\left(e^{j\omega'}\right) = H\left(e^{j\omega'}\right)X\left(e^{j\omega'L}\right) \tag{2.4}$$

$$Y\left(e^{j\omega\prime}\right) = \begin{cases} CX\left(e^{j\omega'L}\right), & |\omega'| \leq \pi/L \\ 0, & otherwise \end{cases} \tag{2.5}$$

In the time-domain, the interpolated signal *y(m)* is given as

$$y(m) = \sum_{k=-\infty}^{\infty} h(m-k)u(k) \tag{2.6}$$

or $$y(m) = \sum_{k=-\infty}^{\infty} h(m-k)x(k/L) \tag{2.7}$$

From equation (2.7), it can be observed that the output *y(m)* depends on the coefficients of the interpolation filter *h(m)*.The direct form finite impulse response (FIR) structure for the interpolator design has been discussed by Crochiere and Rabiner in [69], but the computational complexity of this structure is not less, as the number of unit delays needed in this structure is *L-1*. Another improved design of the interpolation filter, which removes the drawbacks of direct form FIR structure is the poly-phase filter structure [133], [134]. However, the problem in this filter design for the poly-phase structure is its complexity; as we have to separately design each and every poly-phase filter for the given

set of up-converted input samples. Parks and Kolba [135] proposed the minimum mean square error (MMSE) design of the interpolator, in which the poly-phase filters have been incorporated. But, this design can not be applied to the arbitrary interpolation factors.

Polydoros and Protonotarios put forth an FIR filter design in [136], which also minimizes the mean square interpolation error and which can be used to interpolate the signal at a fractional interval. For the implementation of this structure, the knowledge of autocorrelation function of the input signal is required. Another filter design method is the window technique. However, the various commonly used windows are Hamming, Kaiser and Blackman windows. The Kaiser window as well as Blackman window are the best window techniques for the filter design, because the oscillations in the stop-band decay gradually and the ringing effect at the band edge is minimal. It occurs at the expense of an increase in the width of transition band of the filter. A more sophisticated design technique than the window method is equi-ripple FIR filter [69] design based on the Chebyshev approximation method [132]. This filter is most favourable in the sense that the peak (weighted) approximation error in the frequency-domain over the frequency range of interest is minimized. But at low cut-off frequencies (near to $\omega = 0.05$), the equi-ripple design is not considered as optimum because the required filter order 'N' (impulse response duration) has very high values. The half-band filter [113], [133] is an exceptional case of the FIR filter having equal pass-band and stop-band ripple factor values, and the sum of pass-band and stop-band edge frequencies being half of the bandwidth. For an interpolator having large interpolation factor, the cascading of multiple half-band filters guides to the computationally efficient design. For the arbitrary up-sampling factor, the half-band filter design can not be applicable.

Another class of interpolation filter is CIC filter. The CIC filters have been proposed by Hoganauer in [6]. The CIC filters are an unparalleled class of the digital filters that

present a computationally effective manner of implementing the narrowband low-pass filter for anti-aliasing. The CIC filter uses only delays and summation units, and it does not require multiplication operations as in an FIR filter [137]. Occasionally, these are referred to as filters without multipliers. The filter can be implemented by cascading either the integrator with the comb filter or vice-versa. The low-pass frequency response can be improved by cascading '*K*' number of CIC filter stages. These filters also have a linear phase response like the FIR filters. It seems that the CIC filters are quite effective, but it is worth noting that the CIC filters are unstable.

The above discussed filter designs can only be applicable for the integer interpolation factor. To interpolate the signal by a fractional factor, the cascading of interpolation and decimation filter [27], [138] can be utilized. The FIR filters can be employed for the interpolation as well as decimation. In this design, the input signal is first interpolated and then decimated to get the required fractional interpolated output signal. For the large fractional rate factors, this method demands higher order interpolation and decimation filters, which is not practically feasible due to the limitation of hardware resources. Hence, to reduce the resource utilization and computational needs, Farrow [25] has proposed a structure of FIR filters in cascade, known as the Farrow filtering structure. The advantage of this filter is that, it consists of a continuous valued parameter called fractional rate factor, which can be used to interpolate the signal by any arbitrary interpolation factor. The computational needs of this filter are also less than that of cascaded combination of the interpolator and decimator. This filter can be applied to any integer as well as non-integer interpolation factor. In the next section, some filter types have been reviewed, which are predominantly significant and efficient for the FPGA implementation.

2.2 MOVING AVERAGE FILTER

These filters can be realized using the delays, adders/subtractors, but they do not require multipliers, which make the realization of this filter inexpensive. All weights of the moving average filter are set to 1. The figure 2.4 shows six-weight moving average filter.

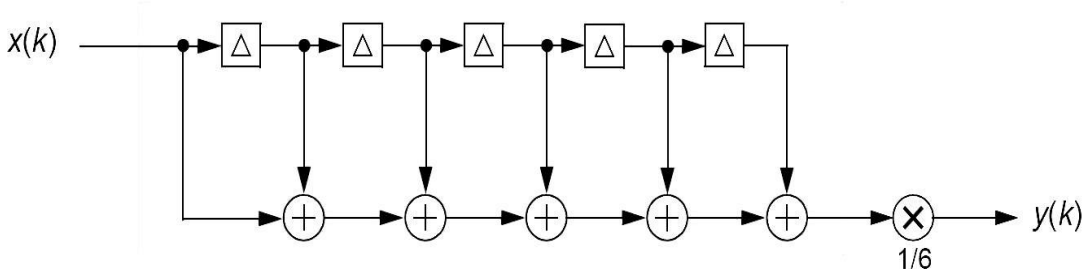

Figure 2.4: Six-weight moving average filter

This filter has simple low-pass characteristic with three spectral zeroes from *0* to $f_s/2$. For this simple moving average filter, the frequency response can be plotted by taking the number of weights (*N*), and then in the range 0 to f_s. There will be *N*-1 evenly spaced spectral zeroes, which mean there are *N*/2 spectral zeroes in the range of *0* to $f_s/2$. Hence for the six-weight filter, we have three evenly spaced spectral zeroes visible, and hence for a sampling rate of 10 MHz, these are 1.66 MHz apart. The figure 2.5 shows the response of a six-weight moving average filter.

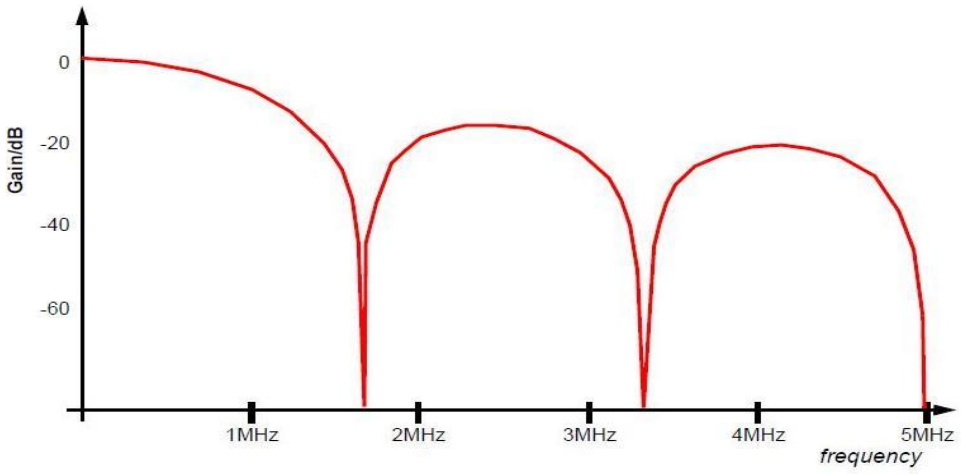

Figure 2.5: Frequency response of a six-weight moving average filter

2.3 DIFFERENTIATOR

A differentiator is a two-weight filter, with values of +1 and -1, which can be shown in figure 2.6.

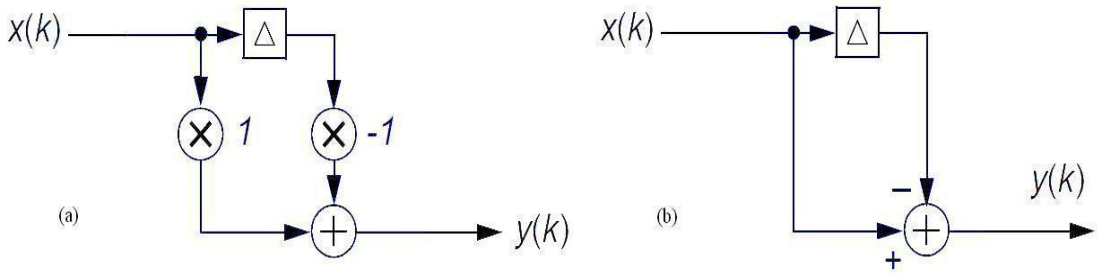

Figure 2.6: (a) Differentiator with weights 1 and -1

(b) Differentiator without multipliers

The figure 2.6 (a) can be shown without the multipliers. The multiplication by +1 can be incorporated as a direct connection to the adder, and multiplication by -1 just requires that the adder is in fact a subtractor as shown in figure 2.6 (b). It has the simple high-pass magnitude response with no multipliers required.

The output may be written as

$$y(k) = x(k) - x(k-1) \tag{2.8}$$

and, the output in *z*-domain may be shown as

$$Y(z) = X(z) - X(z)Z^{-1} \tag{2.9}$$

which can also be written as

$$Y(z) = X(z)[1 - Z^{-1}] \tag{2.10}$$

and, hence the differentiator transfer function is

$$H(z) = \frac{Y(z)}{X(z)} = [1 - Z^{-1}] \tag{2.11}$$

2.4 INTEGRATOR

An integrator is basically a single-weight IIR filter, and it may be shown in figure 2.7.

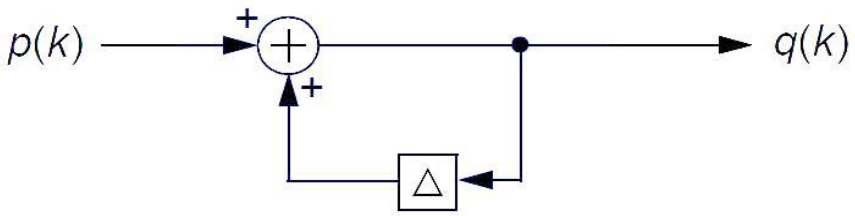

Figure 2.7: Integrator (single-weight IIR filter)

It has the low-pass characteristics and obviously no multipliers are required. The time-domain output may be shown as

$$q(k) = p(k) + q(k-1) \tag{2.12}$$

and, the output in the z- domain may be shown as

$$Q(z) = P(z) + Q(z)Z^{-1} \tag{2.13}$$

It may also be written as

$$Q(z)[1 - Z^{-1}] = P(z) \tag{2.14}$$

and, hence the integrator transfer function is

$$H(z) = \frac{Q(z)}{P(z)} = \frac{1}{1-Z^{-1}} \tag{2.15}$$

2.5 COMB FILTER

The comb filter architecture is similar to that of a differentiator with the modification that the single delay in it has been replaced by the cascading of *N* delays, and all the weights are set to +1 and -1 at either end of the filter. The figure 2.8 shows the eight delay comb filter. By using the *z*-notation to represent the eight delays, the figure 2.8 can also be represented as shown in figure 2.9. A comb filter with *N* sample delays (or *N+1* weights) will be having *N* evenly spaced spectral zeroes from 0 to $f_s/2$. As a result, when the sample rate is set to f_s = 10 MHz, the eight delay comb filter shown in figure 2.9 will have four spectral zeroes from 0 to 5 MHz, at the spacing of 1.25 MHz, which have been demonstrated in figure 2.10.

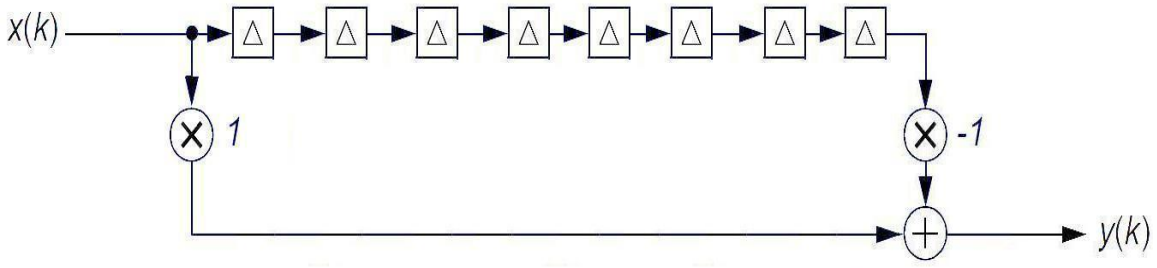

Figure 2.8: A comb filter having 8 delay elements

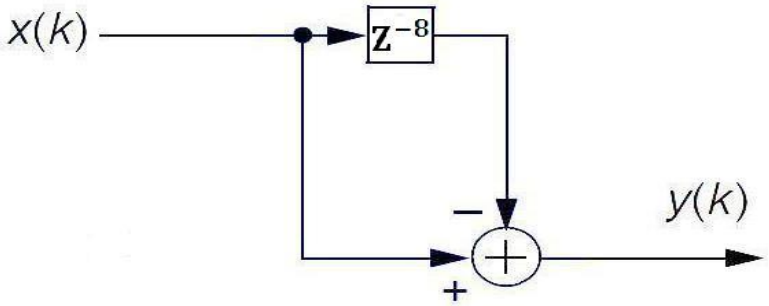

Figure 2.9: Equivalent structure of figure 2.8

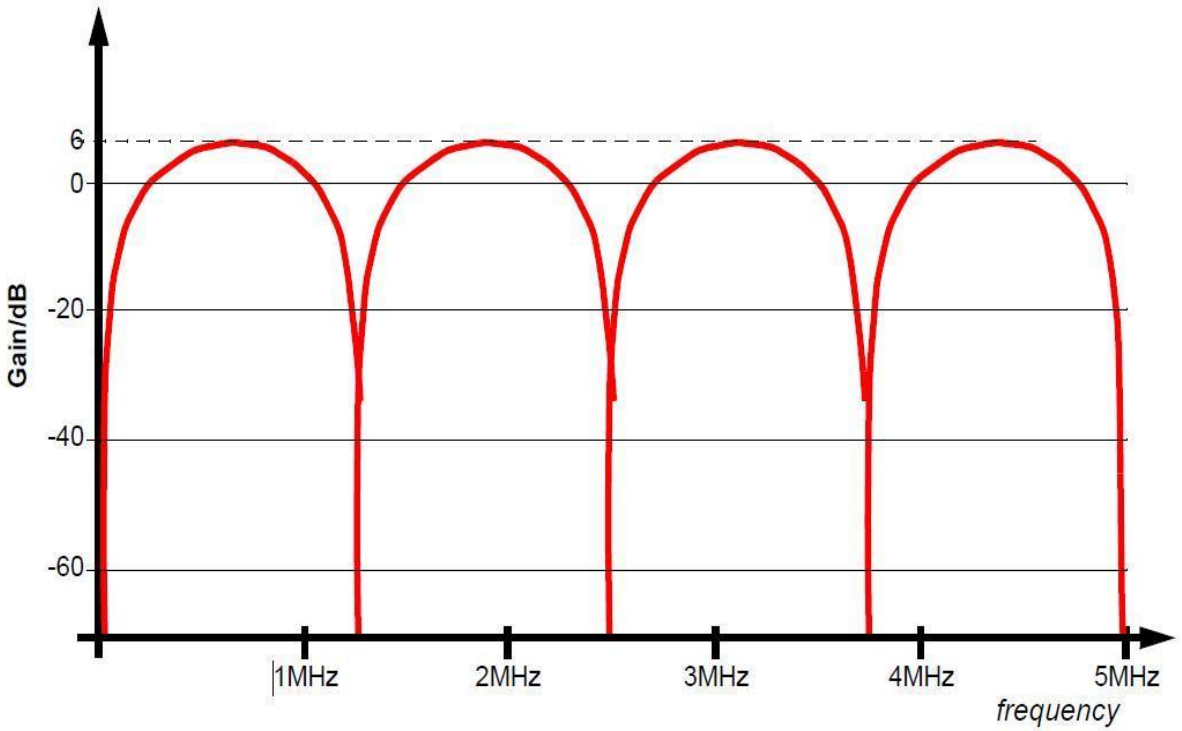

Figure 2.10: Frequency response of eight-weight comb filter

2.6 INTEGRATOR COMB FILTER

An integrator comb structure can be used to implement the moving average filter with the combination of the integrator and comb filter mentioned in sections 2.4 and 2.5. For the eight-weight filter structure

$$H(z) = \left(\frac{1}{1-Z^{-1}}\right)(1 - Z^{-8})\frac{1}{8} \tag{2.16}$$

It may be written as

$$H(z) = \left(\frac{1-Z^{-8}}{1-Z^{-1}}\right)\frac{1}{8} \tag{2.17}$$

The equation (2.17) can be written as

$$\left(\frac{1-Z^{-8}}{1-Z^{-1}}\right)\frac{1}{8} = (1 + Z^{-1} + Z^{-2} + Z^{-3} + Z^{-4} + Z^{-5} + Z^{-6} + Z^{-7})\frac{1}{8} \tag{2.18}$$

From equation (2.18), it is clear that an integrator and *M*-weight comb is identical to the *M*-1-weight moving average filter. The figure 2.11 depicts an integrator comb structure with eight delays, and figure 2.12 shows reduced form of integrator comb structure with eight delays.

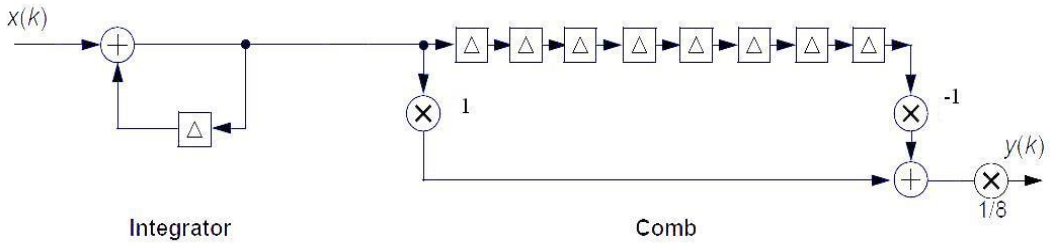

Figure 2.11: Integrator comb structure with 8 delays

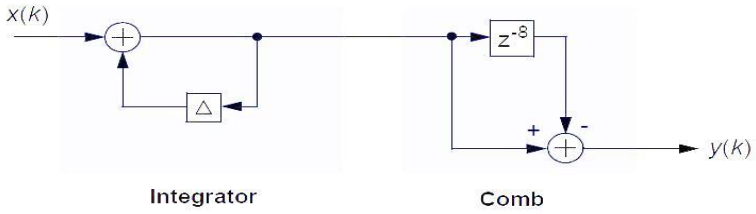

Figure 2.12: Reduced form of integrator comb structure with 8 delays

The figure 2.12 clearly indicates that the integrator comb filter has the advantage that it has only two additions as compared to the 8 additions in the moving average filter.

2.7 CASCADED INTEGRATOR COMB (CIC) FILTER

The CIC filter is a class of hardware-efficient linear phase FIR digital filters. The CIC filters realize sampling rate decrease (decimation) and sampling rate increase (interpolation) without using multipliers. A CIC filter consists of an equal number of stages of the ideal integrator filters and the comb filters. Its frequency response may be tuned by selecting the suitable number of cascaded integrator and comb filter pairs. The extremely symmetric structure of the CIC filter authorizes effective implementation in the hardware. However the disadvantage of a CIC filter is that, its pass-band is not flat, which is disagreeable in many applications. Fortunately, this problem can be alleviated by using a compensation filter. The CIC filter can also be implemented very efficiently in hardware due to its symmetric structure. The CIC decimator would have *N* cascaded integrator stages clocked at f_s, followed by a change in the rate by a factor *R*, followed by *N* cascaded comb stages consecutively running at f_s/R. The figure 2.13 manifests the three-stage decimating CIC filter. The CIC interpolator would have *N* cascaded comb stages running at f_s/R, followed by a zero-stuffer, followed by *N* cascaded integrator stages running at f_s. The figure 2.14 shows the three-stage interpolating CIC filter.

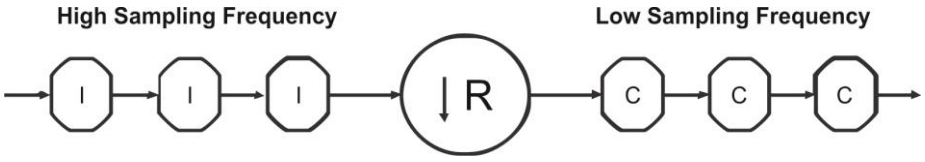

Figure 2.13: Decimating CIC filter (3 stages)

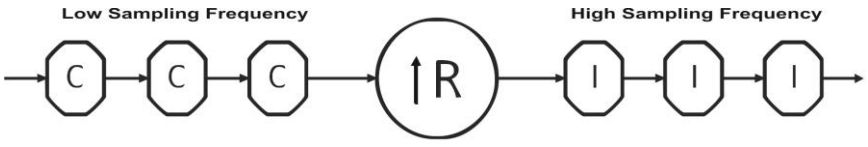

Figure 2.14: Interpolating CIC filter (3 stages)

The transfer function of the CIC filter [6] in z-domain is given as

$$H(z) = \left(\frac{1-z^{-RM}}{1-z^{-1}}\right)^N \tag{2.19}$$

In equation (2.19), R is the decimation or interpolation ratio, *M* is the number of samples per stage (usually it is taken as 1) and *N* is the order of the CIC filter. The numerator $(1 - z^{-RM})^N$ represents the transfer function of a differentiator and the denominator $(1 - z^{-1})^N$ indicates the transfer function of an integrator. A very poor magnitude response of the comb filter is improved by cascading the several identical comb filters. The transfer function *H (z)* of the multi-stage comb filter composed of *K* identical single-stage comb filters is given as

$$H(z) = \left(\frac{1}{N}\frac{1-z^{-N}}{1-z^{-1}}\right)^K \tag{2.20}$$

The figure 2.15 exemplifies that how the multi-stage realization improves the selectivity and the stop-band attenuation of the overall filter. The selectivity and stop-band attenuation are augmented with the increasing number of comb filter sections [139], [140]. The filter has multiple nulls with multiplicity equal to the number of sections. Consequently, the stop-band attenuation in the null intervals is very high. The figure 2.16 illustrates a monotonic decrease of the magnitude response in the pass-band, called the pass-band droop.

2.8 CIC FILTERS IN DECIMATION AND INTERPOLATION

The CIC filters are utilized in the multi-rate systems for constructing the efficient decimators and interpolators. The comb filter's capacity to perform filtering without multiplications is highly admirable to be applied to the exalted rate signals. Moreover, the CIC filters are suitable for the large conversion factors because the low-pass bandwidth is very small. In the multi-stage decimators with the large conversion factor, the comb filter sounds to be the best solution for the first decimation stage. However in the interpolators, the comb filter is convenient for the last interpolation stage. The multi-rate application of the comb filters was first proposed by Hogenauer in [6] and since that time, the well-known Hogenauer filters have influenced many researchers and practicing engineers.

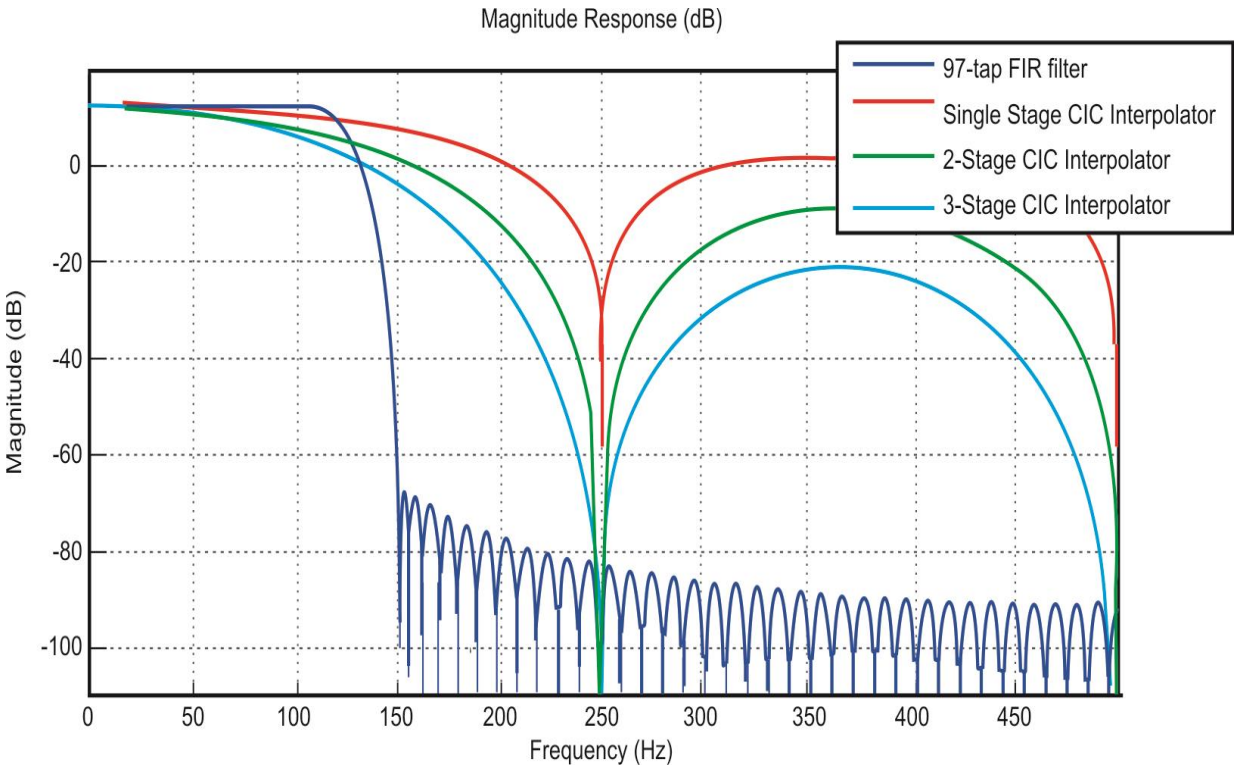

Figure 2.15: Multi-stage CIC filter gain response

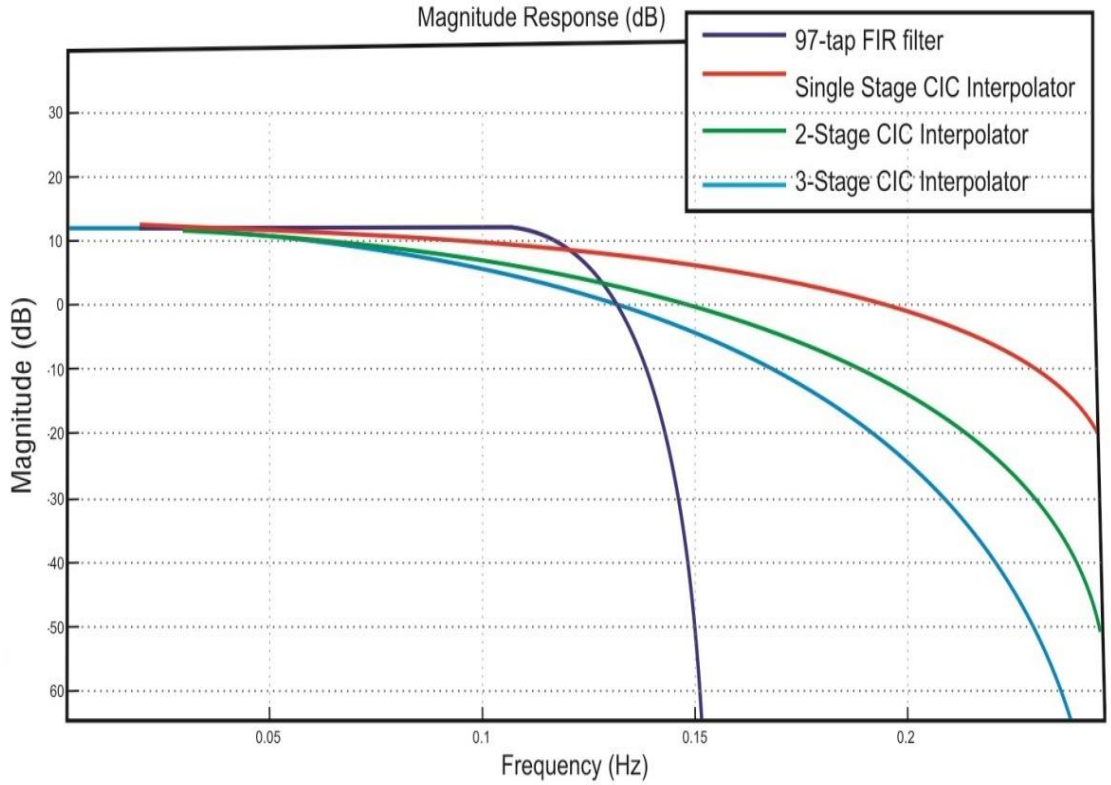

Figure 2.16: Expanded view of the multi-stage CIC filter gain response

2.9 COMPENSATION OF CIC FILTER

A CIC filter can be used as the first stage in decimation when the overall conversion ratio *M* is factorable as

$$M = N \times R \tag{2.21}$$

The overall factor-of-*M* sampling rate conversion system can be implemented by cascading a factor-of-*N* CIC decimator and a factor-of-*R* FIR decimator as shown in figure 2.17(a). The corresponding single-stage equivalent is given in figure 2.17(b). The interpolator with a conversion factor *L* is factorable as

$$L = R \times N \tag{2.22}$$

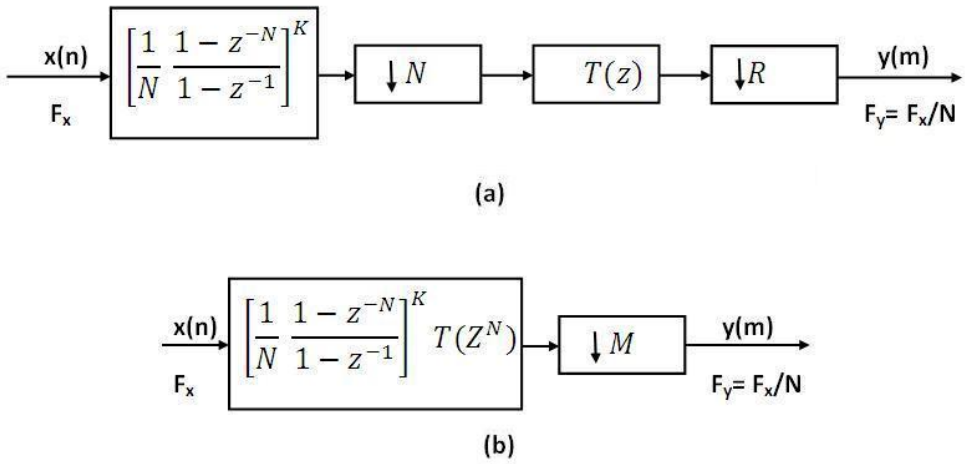

Figure 2.17: Two-stage decimator composed of a CIC filter and an FIR filter (a) Cascade implementation. (b) Single-stage equivalent

It might be beneficial to implement the second (last) stage as a CIC interpolator. The first stage is generally implemented as an FIR filter. The figure 2.18 (a) depicts the two-stage interpolator comprising of the cascade of a factor-of-*R* FIR interpolator and a factor-of-*N* CIC interpolator. The equivalent single-stage is given in figure 2.18 (b). In the two-stage solutions of figure 2.17 and figure 2.18, the role of the CIC decimator (interpolator) is to convert the sampling rate by the large conversion factor *N*. Also, the FIR filter *T(z)*

provides the desired transition band of the overall decimator/interpolator and compensates the pass-band characteristic of the CIC filter [106], [133].

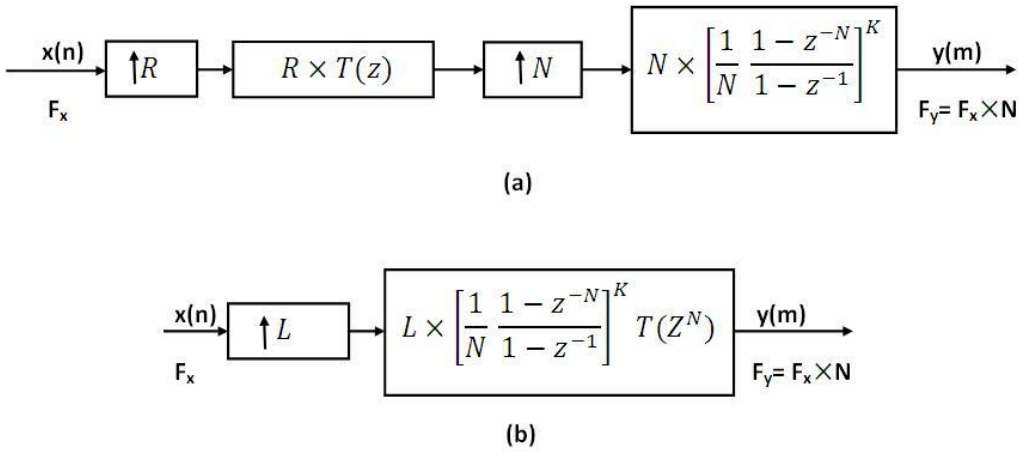

Figure 2.18: Two-stage interpolator composed of an FIR filter in the first stage, and the CIC filter in the second stage

(a) Cascade implementation (b) Single-stage equivalent

The filter $T(z^N)$ ensures that the desired transition band, which compensates the pass-band droop of the comb filter at the first stage. The CIC filter $H(z)$ has its two nulls just in the unwanted pass-bands of the periodic filter $T(z^N)$ that ensure the desired stop-band attenuation of the target two-stage decimator. Ultimately, the frequency response of the overall two-stage decimation filter can be computed as

$$H_1(z) = H(z).T(z^N) \tag{2.23}$$

The figure 2.19 demonstrates the gain responses of CIC filter, compensating filter and the resulting filter. The figure 2.20 displays the pass-band droop of CIC filter, compensating filter and the resulting filter. It is apparent from the above results that the pass-band droop of the CIC filter can be improved by using a FIR filter in cascade with the CIC filter. Thus, the FIR filter compensates the droop of CIC filter. Therefore, this process is known as the compensation of CIC filters.

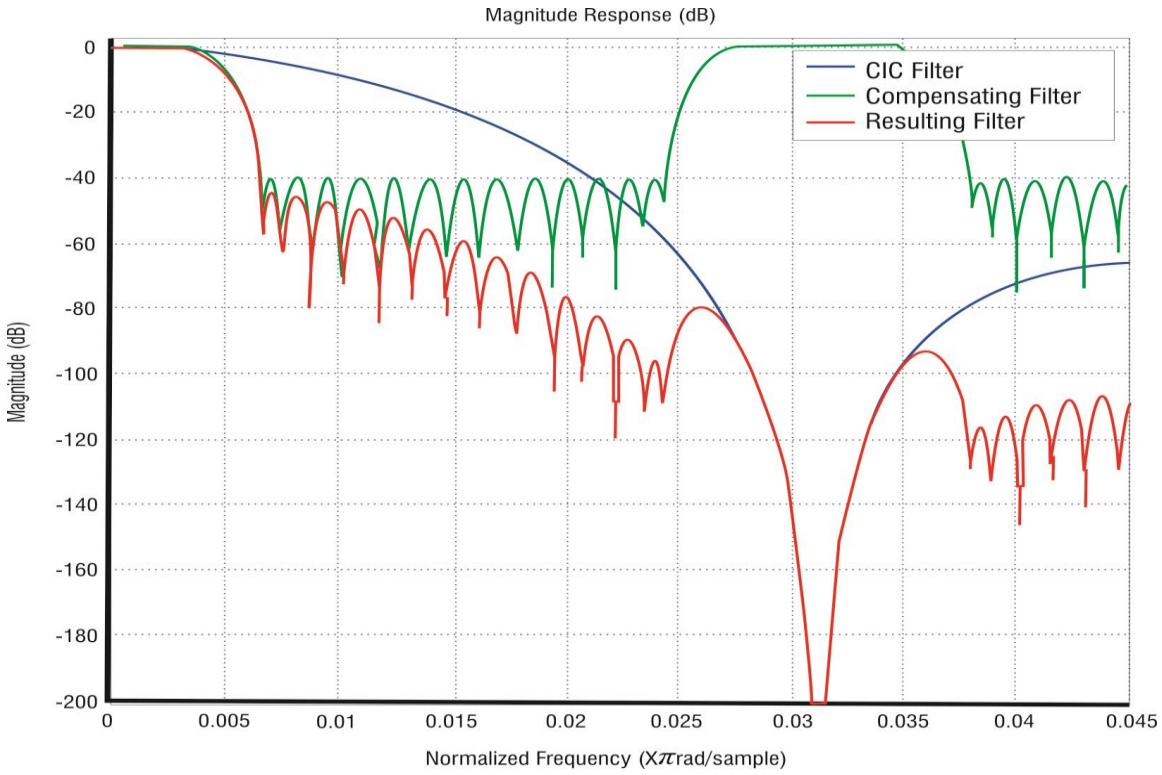

Figure 2.19: Gain responses of the CIC filter, compensating filter and the resulting filter

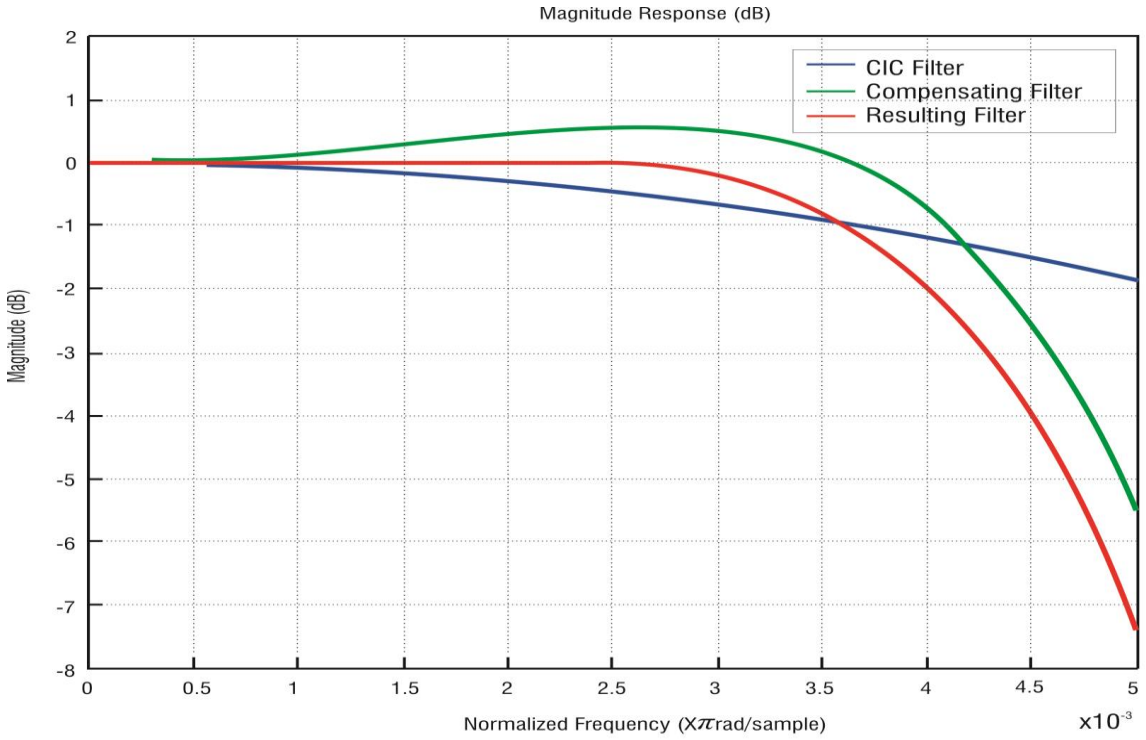

Figure 2.20: Expanded view of the gain responses of the CIC filter, compensating filter and the resulting filter

2.10 FILTERS WITH NON-INTEGER DECIMATION FACTOR

When the decimation factor *1/R* or the interpolation factor *R* is an integer value, the conversion of sampling rate can be conducted conveniently with the assistance of fixed digital filters [7]. In this scenario, where the factors are irrational, it would be impossible to use the fixed digital filters directly. Moreover, if *R* is considered as the ratio of two relatively large prime integers, then in case of the conventional poly-phase implementation, it is quite essential that the order of required filter become enormous. In a nutshell, it means that a large number of coefficients need to be stored in the coefficient memory. In sampling rate conversion by the non-integer factor, it is required to establish the values between existing samples. In this case, it is very convenient to use interpolation filters. Here, the polynomial based filters are generally assumed to provide an efficient implementation form, directly in the digital domain. Such filters witness an effective implementation through the Farrow structure or its higher version [12], [36]. The main advantage of the Farrow structure is based on the presence of fixed FIR filters as one of its ingredients. Eventually, there is only one changeable parameter, called fractional interval μ. Besides this, the control of μ is easier during the operation than in the corresponding coefficient memory implementations [25], and the concept of arithmetic preciseness, not the memory size confines the resolution of μ. These attributes of the Farrow structure make it a very attractive and suitable structure to be implemented using a VLSI circuit or a signal processor.

In figure 2.21, the dashed line separates the filter into a section running at the input signal's sampling-rate and a section running at the output sampling-rate. Note that the output is relabeled to be *y(m)* rather than *y(n)*. This is due to the dissimilar input and output rates. It is noteworthy that the fractional delay is denoted as β_m, and the output sample value changes as the value of parameter β_m gets altered [9], [10], [75].

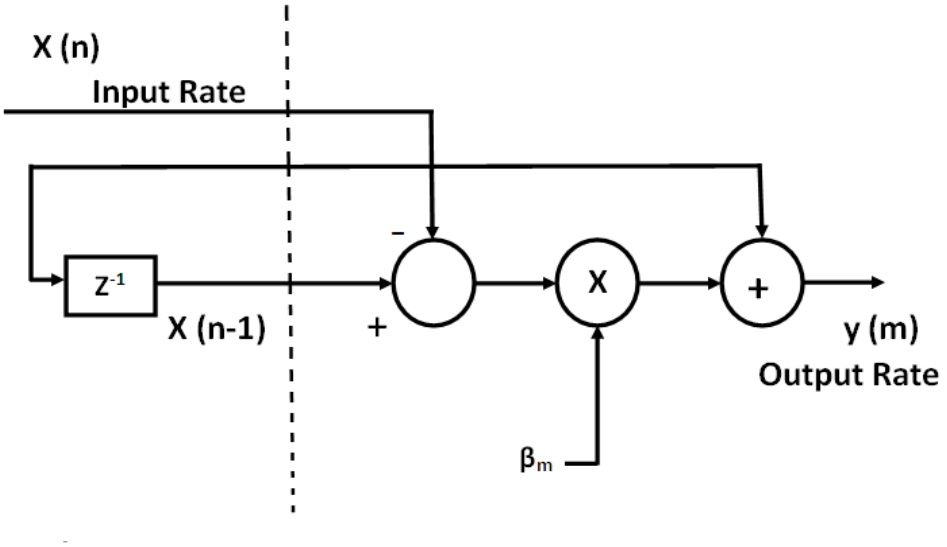

Figure 2.21: Multi-rate Farrow Filter

Now consider a case, where the sampling rate is improved by a factor of *2*. There are two outputs corresponding to every input, therefore the value saved in the delay register will be used twice. For the first input value, the parameter β_m will take on the value 0.5 and the output will be computed as

$$y[m] = 0.5(x[n-1] - x[n]) + x[n] = 0.5x[n-1] + 0.5x[n] \quad (2.24)$$

Before the input sample changes, one more output sample will be computed. The parameter β_m will take the zero value and the output will simply be

$$y[m+1] = x[n] \quad (2.25)$$

Subsequently, the input sample will change and the parameter β_m will be once again set to 0.5, and so forth. To summarize, when increasing the sampling rate by a factor of two, the parameter β_m will cycle between the values [0.5, 0] twice as fast as the input, generating an output accordingly each time it changes. In a general case, it is simply an issue of evaluating which values β_m must take. The formula is simply defined as

$$\beta_m = \left(\frac{mf_s}{f_s'}\right) mod\ 1 \quad (2.26)$$

where, f_s is the input sampling rate and f_s' is the output sampling rate. In order to perform the non-integer SRC, one can use the Farrow filter structure or its modifications directly.

However in many cases, it becomes more efficient to use the cascaded structures engineered by the modification of the Farrow filtering structure and fixed FIR, or the multi-stage FIR filter [8], [14]. The prime advantage of using the cascaded structures instead of the direct modification of the Farrow filter structure lies in the fact that in case of the joint optimization of two building blocks, the computational complexity to generate (practically) the same filtering performance is significantly reduced. This is because of the following reasons. First, the implementation of a fixed linear phase FIR interpolator is not very costly, as compared to the Farrow filter structure. Second and most important fact is that, the requirements for implementing the modification of the Farrow filter structure become significantly milder. It is possible because the FIR filter takes care of the pass-band and stop-band shaping, whereas the Farrow filtering based structure takes care of attenuating images of the FIR filter.

The fractional sampling rate is the requirement of all modern communication systems. The Farrow filter structure is an efficient structure to implement the interpolation filter for the fractional rate change as well as for any arbitrary rate change factor. In this research work, the Farrow filters have been implemented for the fractional delay and arbitrary change in the sample rate conversion on the FPGA platform, which is the basic necessity of any modern digital communication system. In this research work, the CIC filter with and without compensation technique is implemented on the FPGA platform. The Farrow filters are also implemented for the fractional delay and arbitrary change in the sample rate conversion. Both these filters provide better performance than the common filter structures in terms of the speed of operation, cost and power consumption in the real-time. These filters are implemented on the Altera Stratix-II-EP2S15F484C3 FPGA and simulated with the help of Quartus II v9.1sp2.

2.11 DESIGN EXAMPLES OF THE MULTI-STAGE CIC FILTER

This subsection illustrates the simulation results of the DUC filter design for various wireless standards using the multi-stage CIC filters without compensation techniques.

2.11.1 Simulation Results of the DUC Filter Design for Global System for Mobile Communication (GSM)

The specifications for the DUC Filter for GSM are as follows

- Input Sampling Frequency : 34.667 KHz
- Output Sampling Frequency: 270.83 KHz
- Pass-band Edge : 0.08 MHz
- Stop-band Edge : 0.1 MHz
- Pass-band Ripple : 0.1 dB
- Stop-band Attenuation : 65 dB

Using the above specifications, the multi-stage filter is implemented. The response of the filter is shown in figure 2.22, and its simulation results are plotted in figure 2.23.

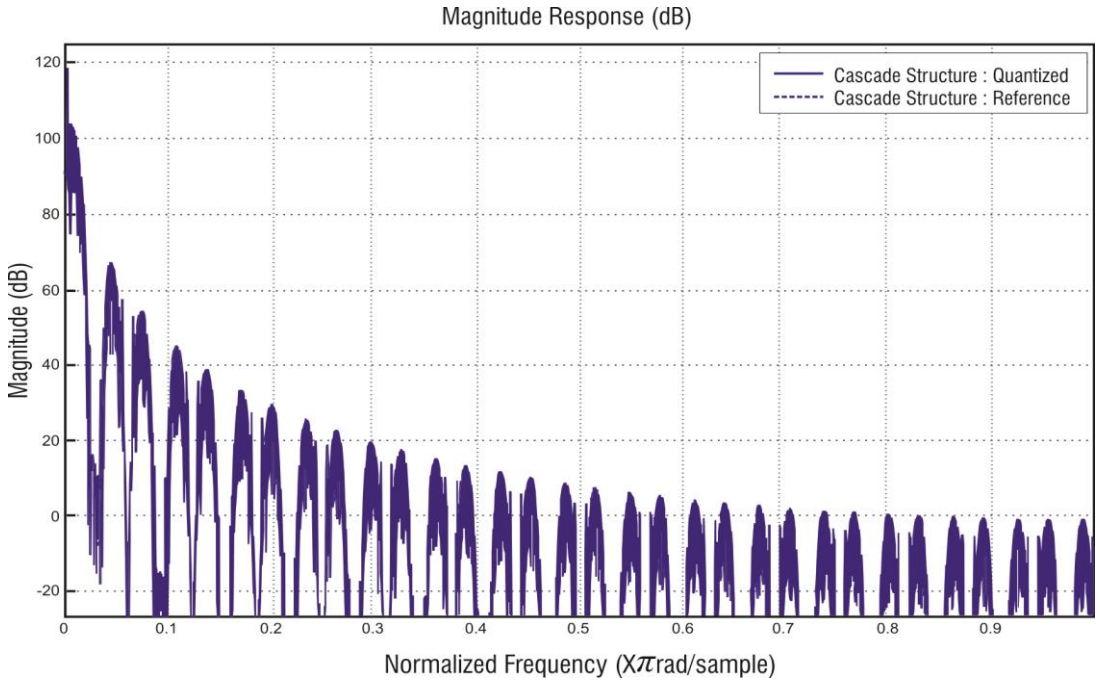

Figure 2.22: Magnitude response of the DUC filter for GSM

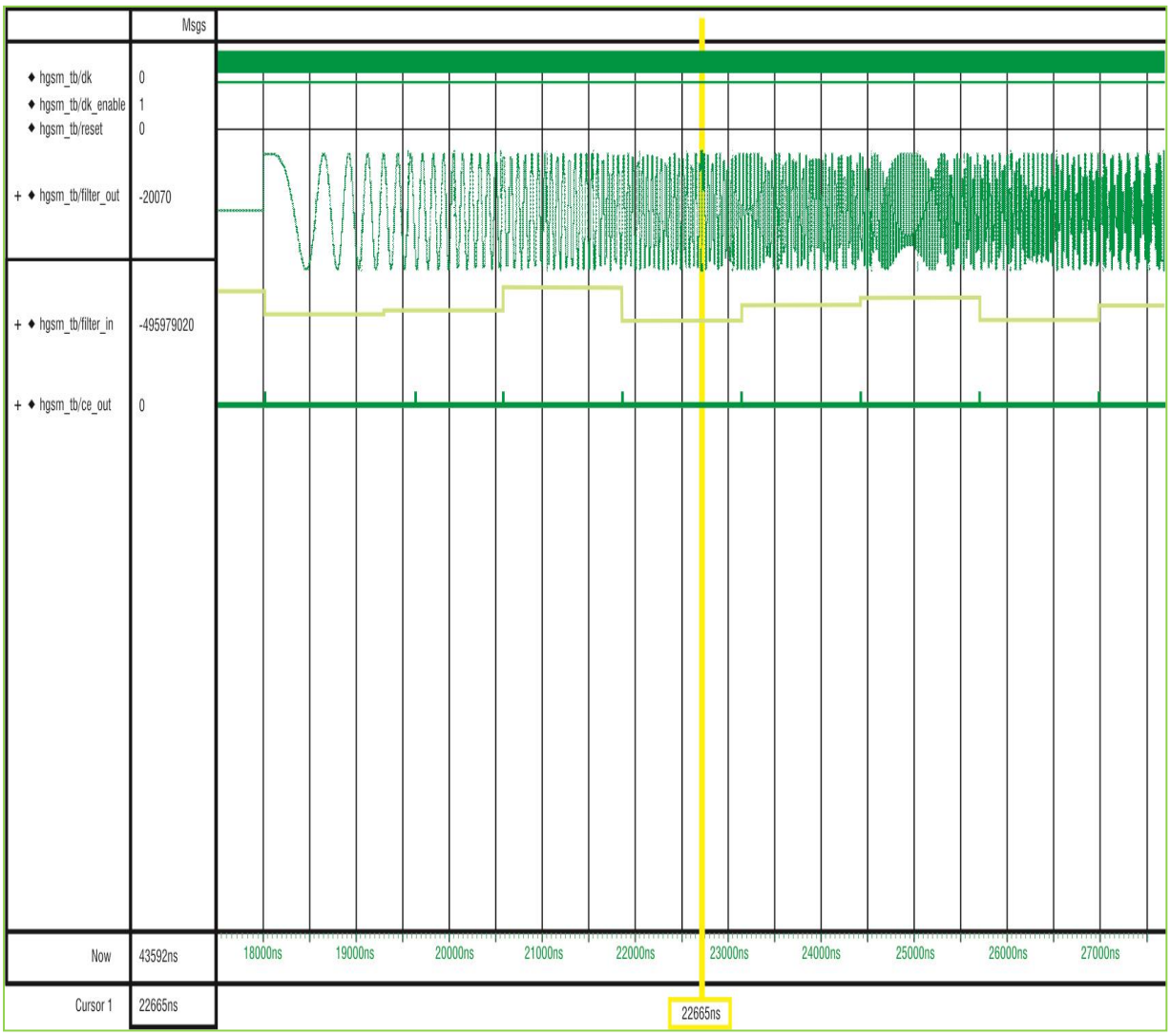

Figure 2.23: Modelsim simulation of the DUC filter design for GSM

2.11.2 Simulation Results of the DUC Filter Design for Wideband Code Division Multiple Access (WCDMA)

The specifications for the DUC Filter for WCDMA are as follows

- Input Sampling Frequency : 3.84 MHz
- Output Sampling Frequency: 61.44 MHz
- Pass-band Edge : 2 MHz
- Stop-band Edge : 2.5 MHz
- Pass-band Ripple : 0.5 dB
- Stop-band Attenuation : 44 dB

Using the above specifications, the multi-stage filter is implemented. The response of the filter is shown in figure 2.24, and its simulation has been presented in figure 2.25.

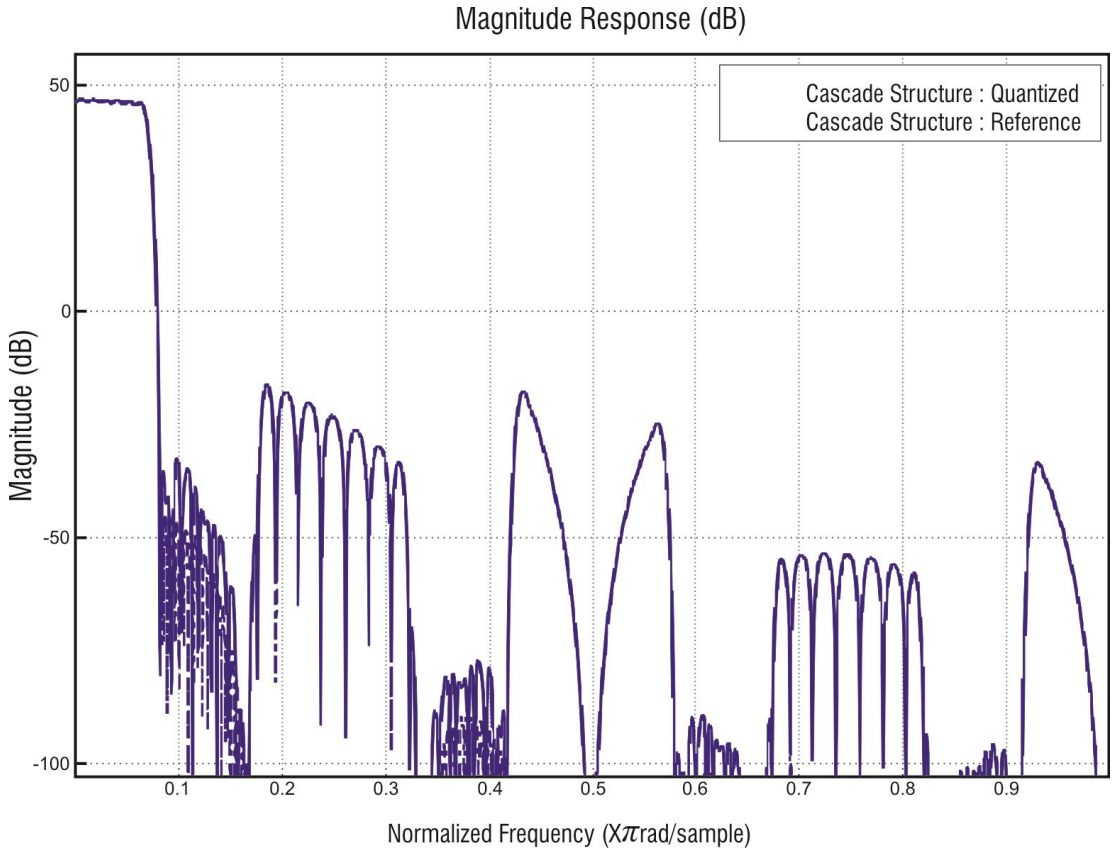

Figure 2.24: Magnitude response of the DUC filter for WCDMA

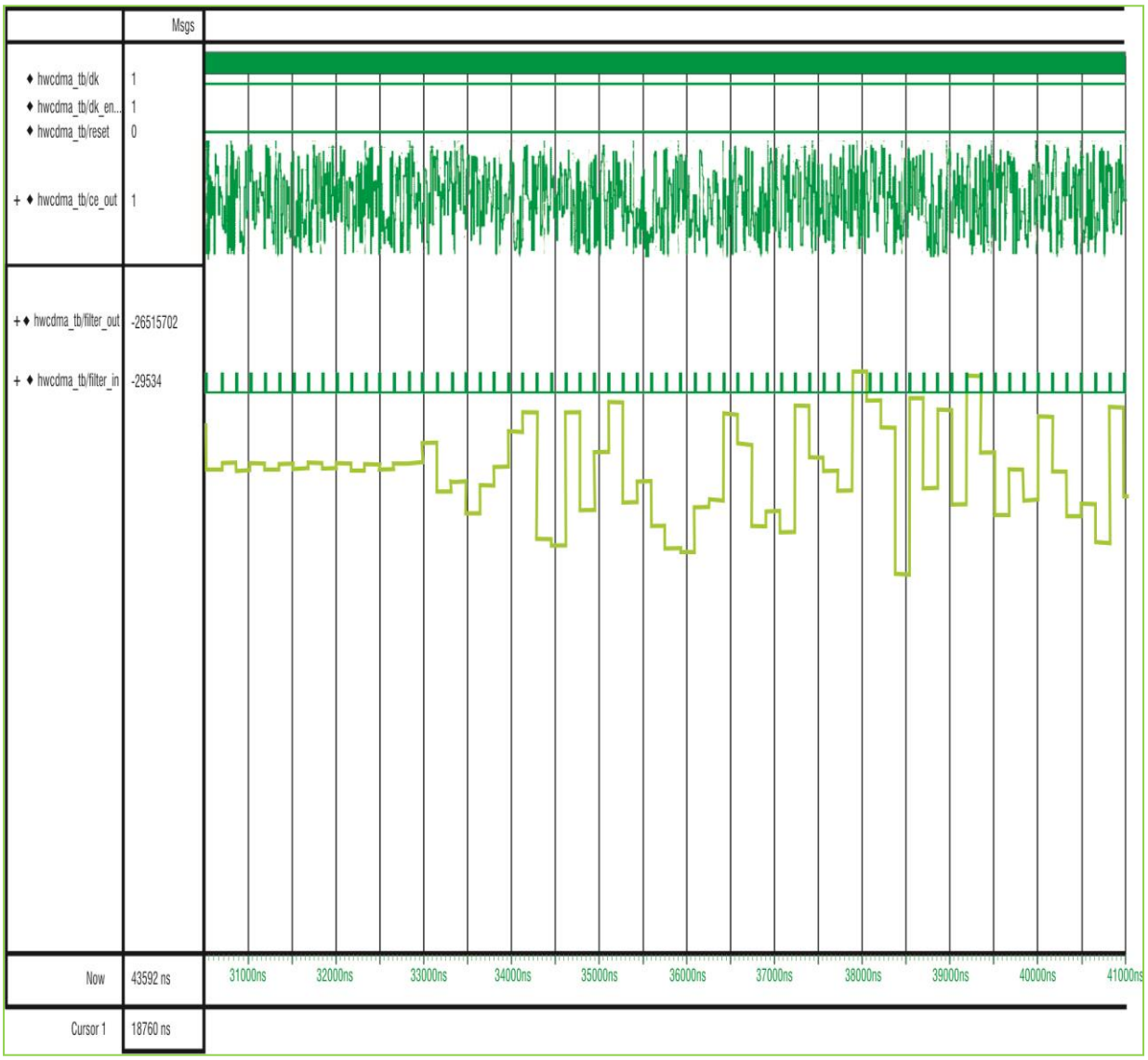

Figure 2.25: Modelsim simulation of the DUC filter design for WCDMA

2.11.3 Simulation Results of the DUC Filter Design for Multichannel Worldwide Interoperability for Microwave Access (WiMax)

The specifications for DUC Filter for multichannel WiMax are as given below

- Input Sampling Frequency: 16.704 MHz
- Output Sampling Frequency: 133.632 MHz
- Pass-band Edge : 8 MHz
- Stop-band Edge : 10 MHz
- Pass-band Ripple : 0.5 dB
- Stop-band Attenuation : 39 dB

Using the above specifications, the multi-stage filter is implemented. The response of the filter is shown in figure 2.26, and its simulation is depicted in figure 2.27.

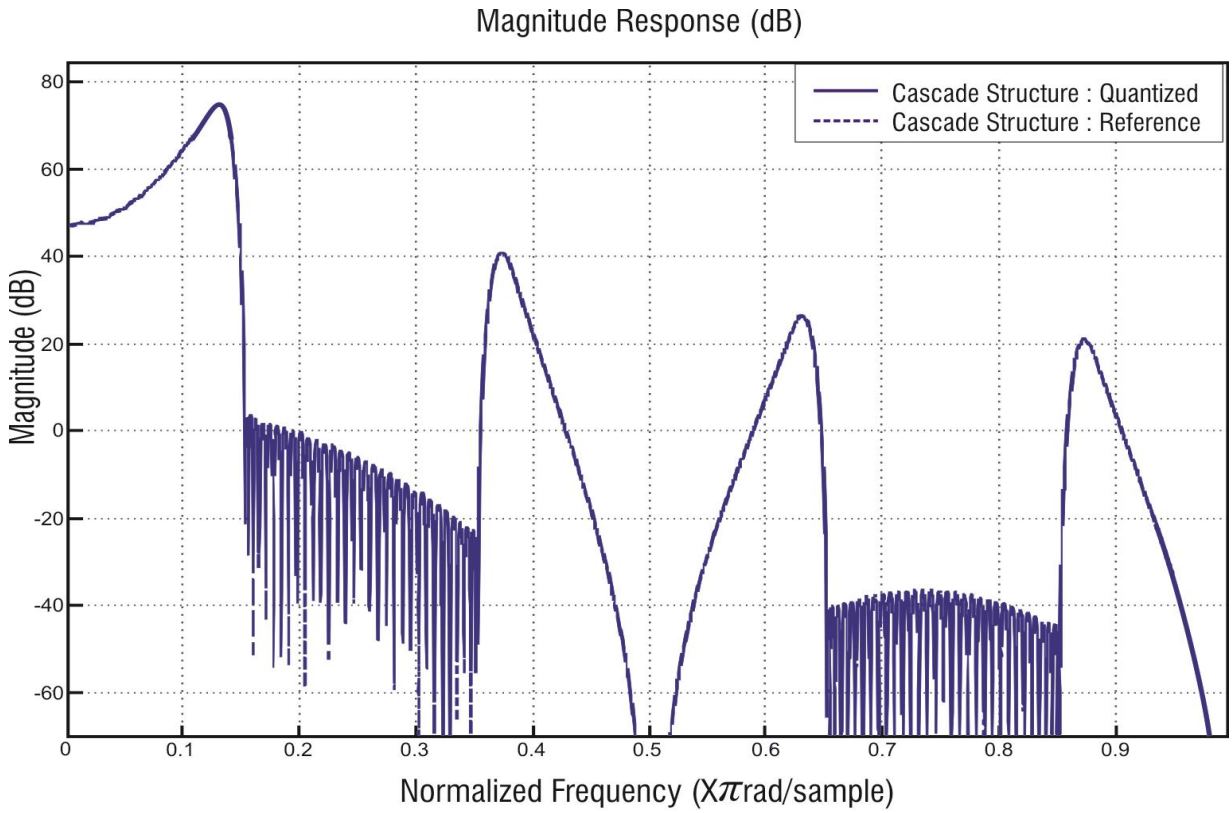

Figure 2.26: Magnitude response of the DUC filter for WiMax

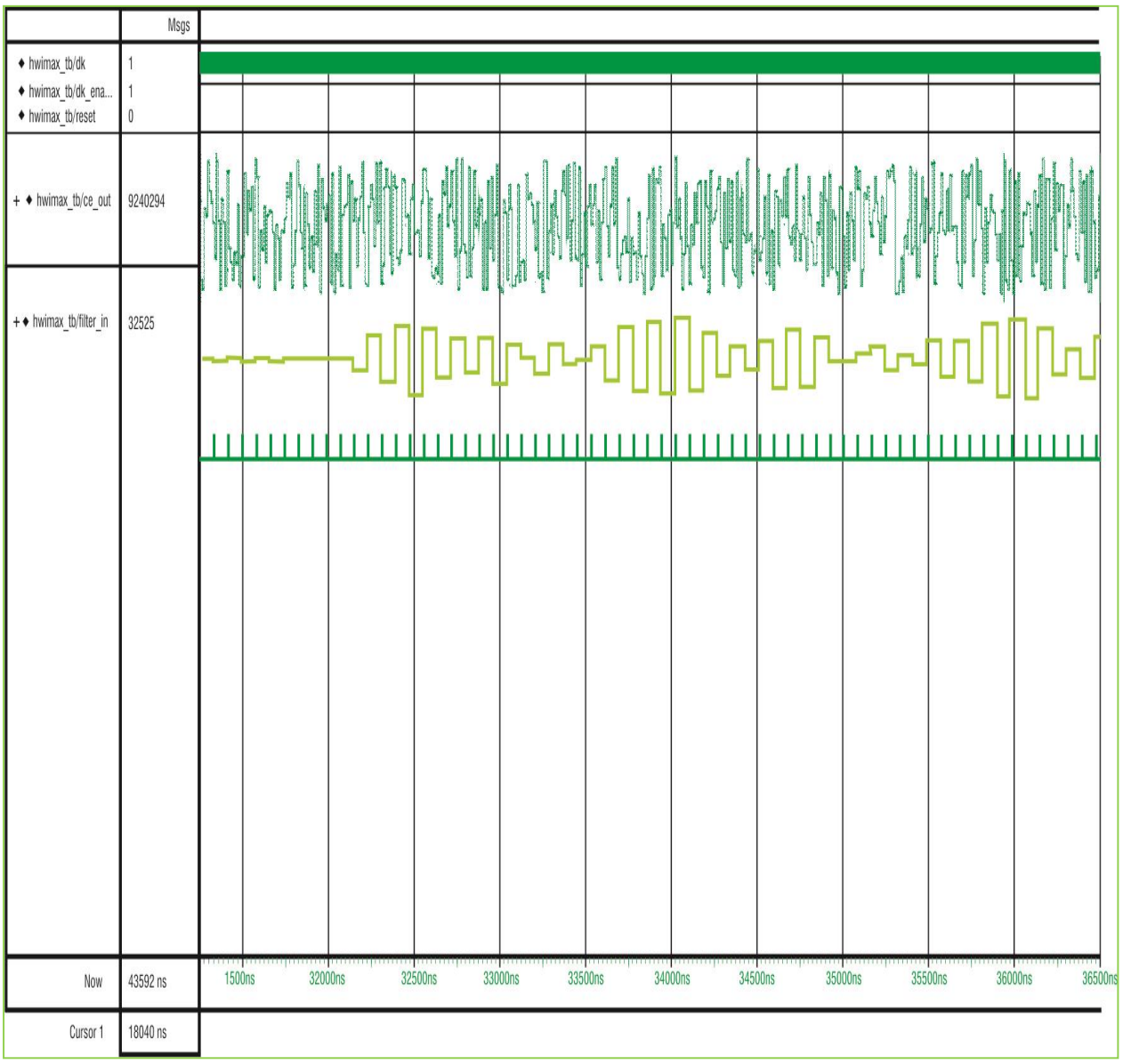

Figure 2.27: Modelsim simulation of the DUC filter design for WiMax

TABLE 2.1
COMPARISON OF IMPLEMENTATION COST AND SPEED ANALYSIS OF THE DUC FILTERS FOR VARIOUS WIRELESS STANDARDS

Property	GSM	WCDMA	WiMAX
Logic Utilization	93%	43%	51%
Worst Case Set up time (t_{su})	20.89ns	20.58ns	7.50ns
Worst Case Clock to Output time (t_{cq})	6.66ns	7.01ns	6.79ns
Clock Frequency	48.82 MHz	39.23 MHz	25.09 MHz

The simulation and implementation results for the different wireless standards are presented in the above sections. From the results presented in table 2.1, it can be concluded that the CIC filters are efficient for the low-cost applications. Due to the absence of multipliers, they also exhibit faster response. But, the pass-band droop present in the CIC filters restrict the scope of their applications. By employing compensation, the response of the CIC filter in the pass-band can be improved, which has been shown in the next section.

2.11.4 Comparison of the CIC Filters With and Without Compensation

The specifications for the DUC for WiMax are as follows

- Input Sampling Frequency : 11.424 MHz
- Output Sampling Frequency: 91.392 MHz
- Pass-band Edge : 4.75 MHz
- Pass-band Ripple : 0.14 dB
- Stop-band Attenuation : 92 dB

Simulation results of the CIC filters are as follows

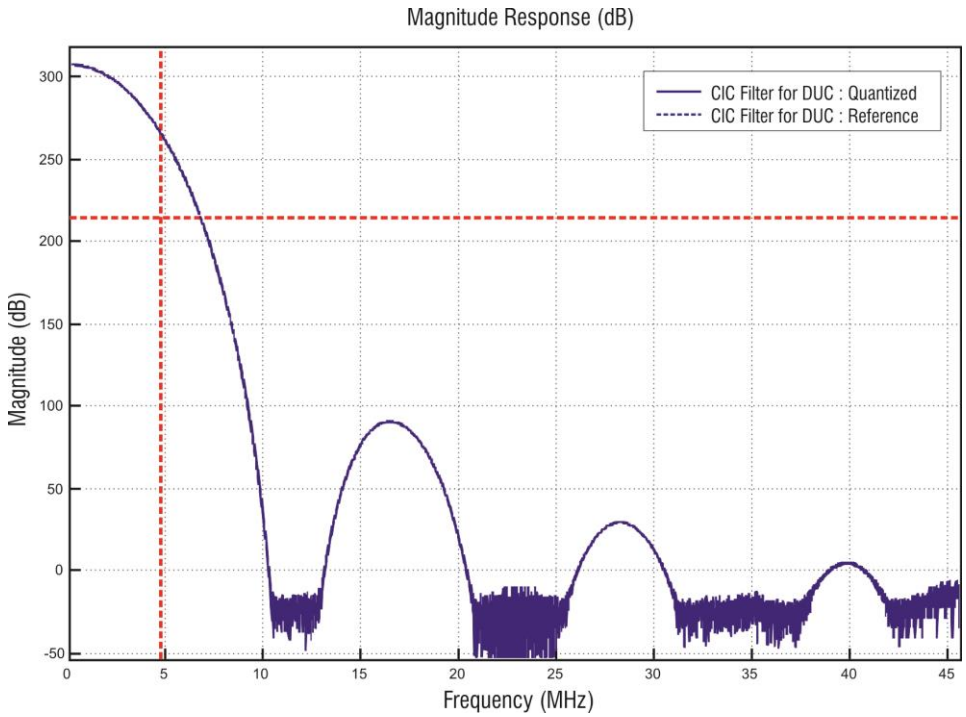

Figure 2.28: Magnitude response of the DUC for WiMax using the CIC filters

Simulation results of CIC filter with compensation is as follows

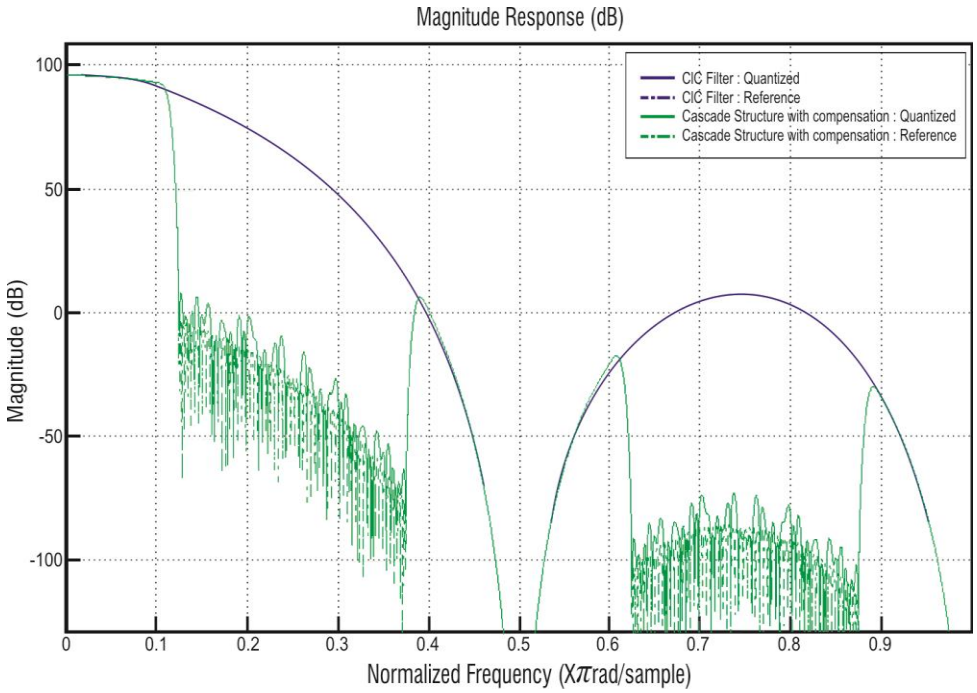

Figure 2.29: Magnitude response of the DUC for WiMax using the CIC filters using compensation

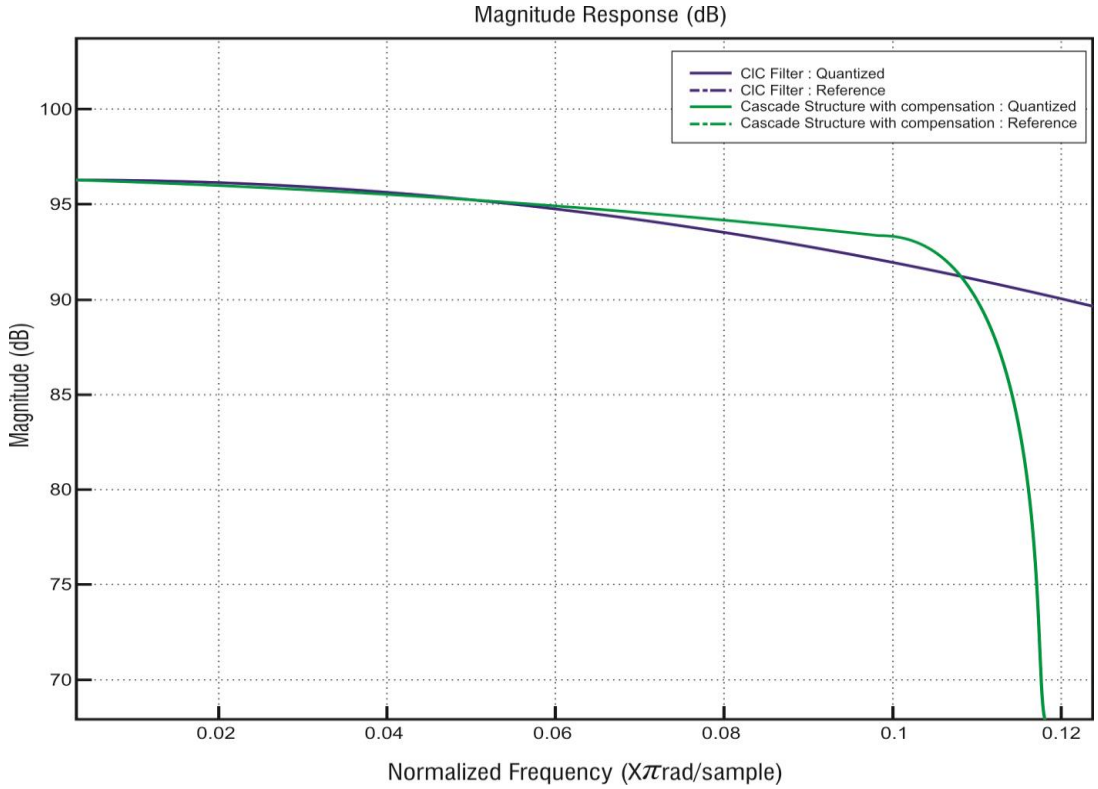

Figure 2.30: Magnitude response of the DUC for WiMax using the CIC filters using compensation: expanded view

TABLE 2.2
COMPARISON OF IMPLEMENTATION COST AND SPEED ANALYSIS OF THE CIC FILTERS AND ITS CASCADED STRUCTURE WITH AND WITHOUT COMPENSATION

Property	**CIC filter**	**Cascaded CIC filter with compensation**
Logic Utilization	29%	63%
Worst Case Set up time (t_{su})	7.425ns	6.654ns
Worst Case Clock to Output time (t_{cq})	7.379ns	6.619ns
Clock Frequency	203.54 MHz	21.91MHz

From figures 2.28, 2.29, 2.30 and table 2.2, it can be established that the pass-band droop present in the CIC filters can be alleviated by employing the compensation technique, but at the cost of extra hardware.

2.12 COMPARISON OF THE FIR AND THE FARROW SAMPLE RATE CONVERSIONS

Consider a design example to change the sampling rate by a factor of 1.536 (192/125). The design to change the sample rate by an arbitrary factor is considered. Both filters have been implemented efficiently in the Altera's Stratix II FPGA family with the device number EP2S15F484C3. The results obtained from the simulations are as follows

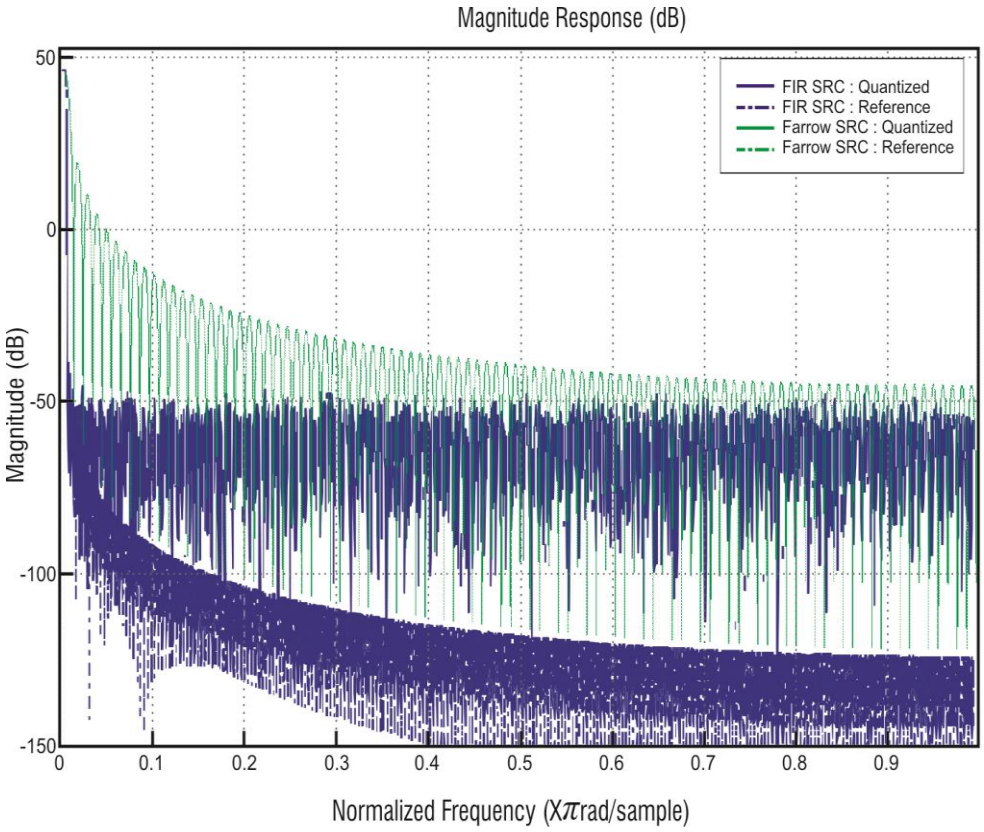

Figure 2.31: Response of the FIR SRC (blue) and Farrow SRC (green) for the factor of 1.536 change in sample rates

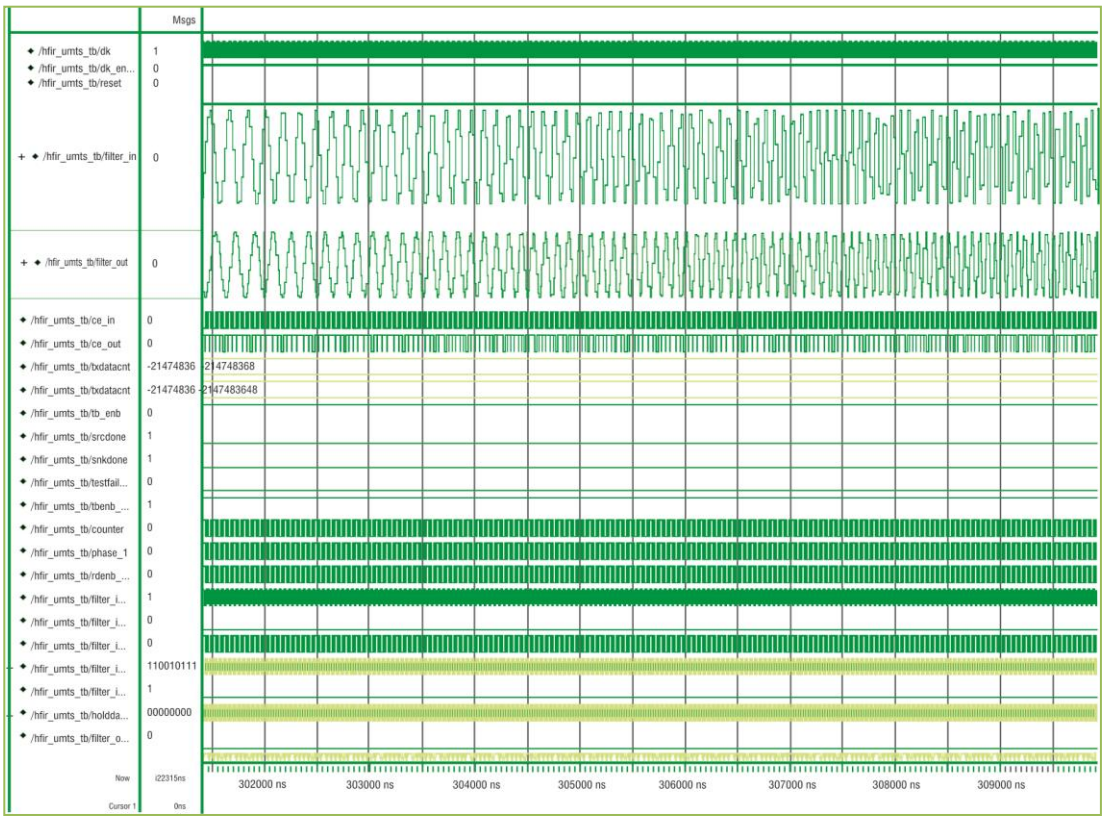

Figure 2.32: Modelsim simulation of the FIR SRC

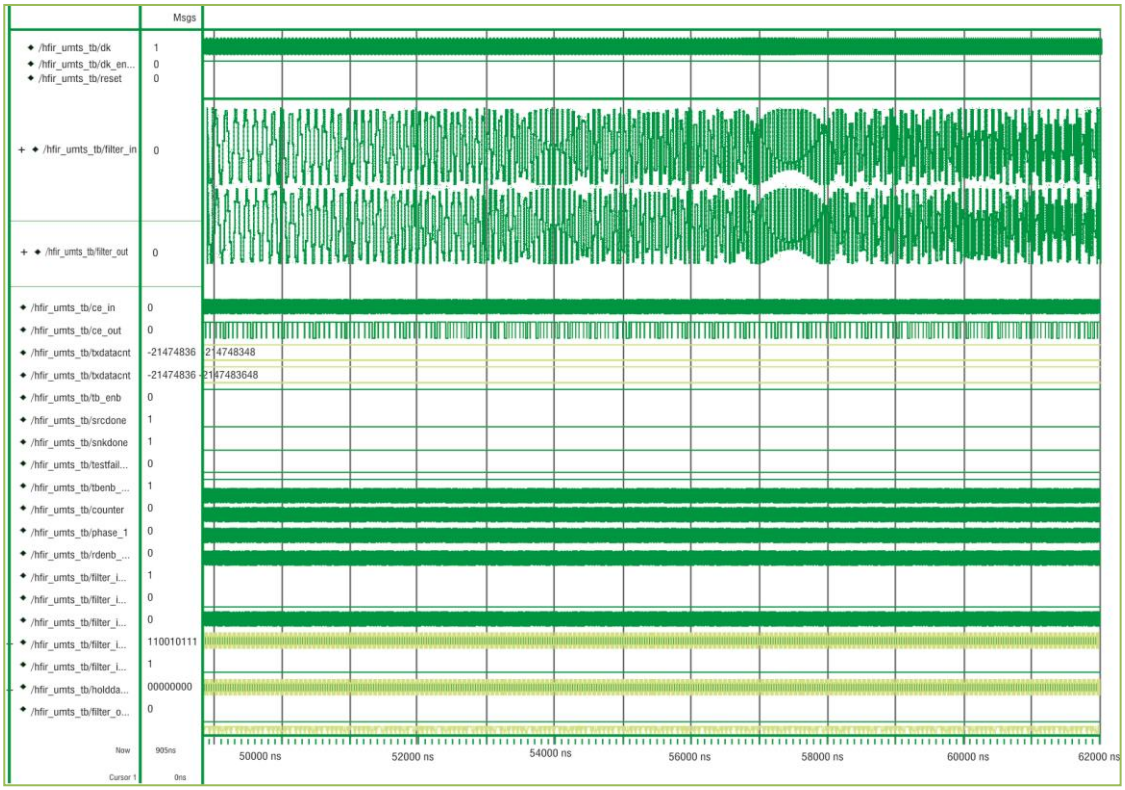

Figure 2.33: Modelsim simulation of the Farow SRC

TABLE 2.3
COMPARISON OF IMPLEMENTATION COST AND SPEED ANALYSIS OF THE FARROW AND THE FIR SAMPLE RATE CONVERSION

Property	Farrow SRC	FIR SRC
Logic Utilization	< 1%	40%
Worst Case Set up time (t_{su})	4.793 ns	7.473 ns
Worst Case Clock to Output time (t_{cq})	7.233 ns	6.944 ns
Clock Frequency	306.18 MHz	49.66 MHz

From figures 2.31, 2.32, 2.33 and Table 2.3, it is apparent that the Farrow filter structure is better than the normal FIR filter. The efficiency of the Farrow filter is more as compared to other one, as the amount of logic utilization required for the coefficients of Farrow filter is much less as compared to the FIR filter. Secondly, the response of the Farrow filter for same SRC is much faster as compared to the FIR SRC. All these sum up to prove that the Farrow filters are more efficient in the implementation as compared to the FIR filter structure.

2.13 DESIGN EXAMPLES OF THE FARROW FILTERS AS SAMPLE RATE CONVERTORS

This subsection illustrates the simulation results of sample rate convertor for GSM and Universal Mobile Telecommunications System (UMTS) wireless standards realized using the Farrow filters.

2.13.1 Simulation Results of the Farrow Filters as Sample Rate Convertor Design for GSM

For the GSM standard, the change of sampling rate is required to be a factor of 1.0832 (677/625). The design to change the sample rate by an arbitrary factor is considered. Using this specification, the Farrow filter for SRC is implemented. The response of the filter is shown in figure 2.34, and its simulation is demonstrated in figure 2.35.

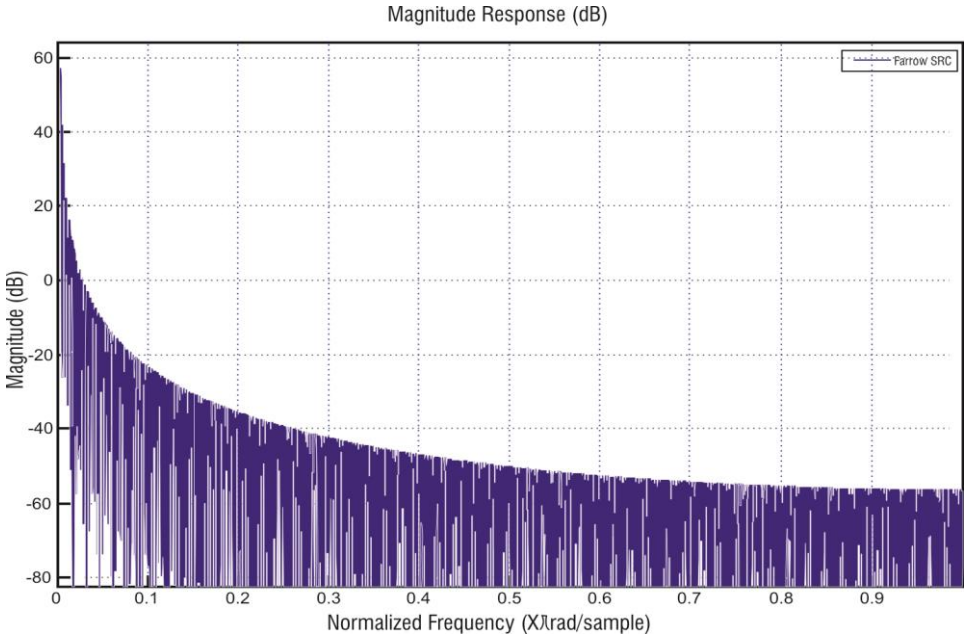

Figure 2.34: Magnitude response of Farrow SRC filter for GSM

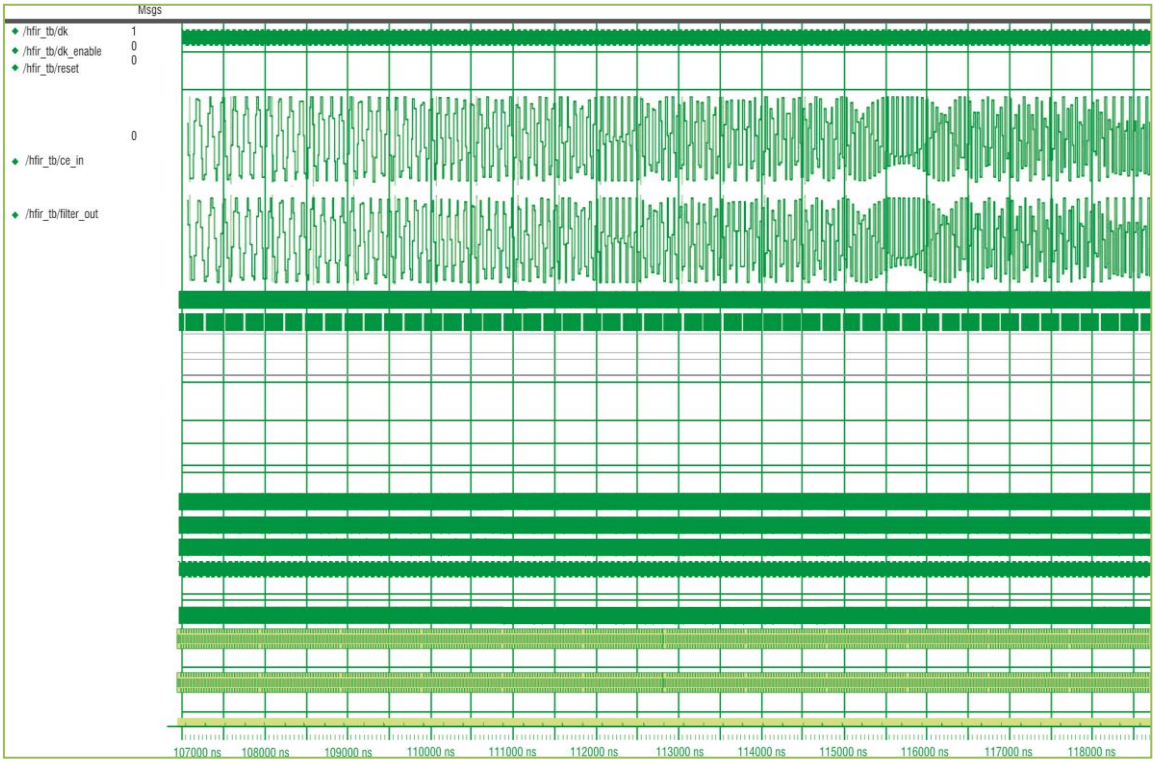

Figure 2.35: Modelsim simulation of Farrow SRC filter design for GSM

2.13.2 Simulation Results of Sample Rate Convertor Design for UMTS

For the UMTS standard, the change of sampling rate is a pre-requisite to be a factor of 1.536 (192/125).The design to change the sample rate by an arbitrary factor is considered. Using this specification, the Farrow filter for the SRC is implemented. The response of filter is shown in figure 2.36, and its simulation is presented in figure 2.37.

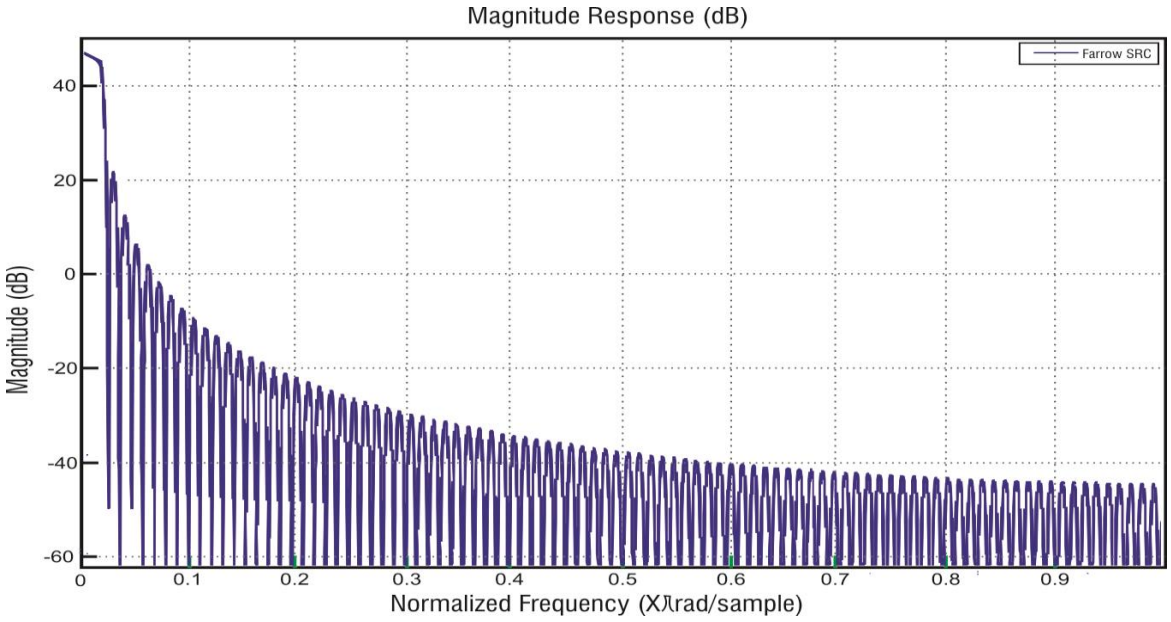

Figure 2.36: Magnitude response of Farrow SRC filter for UMTS

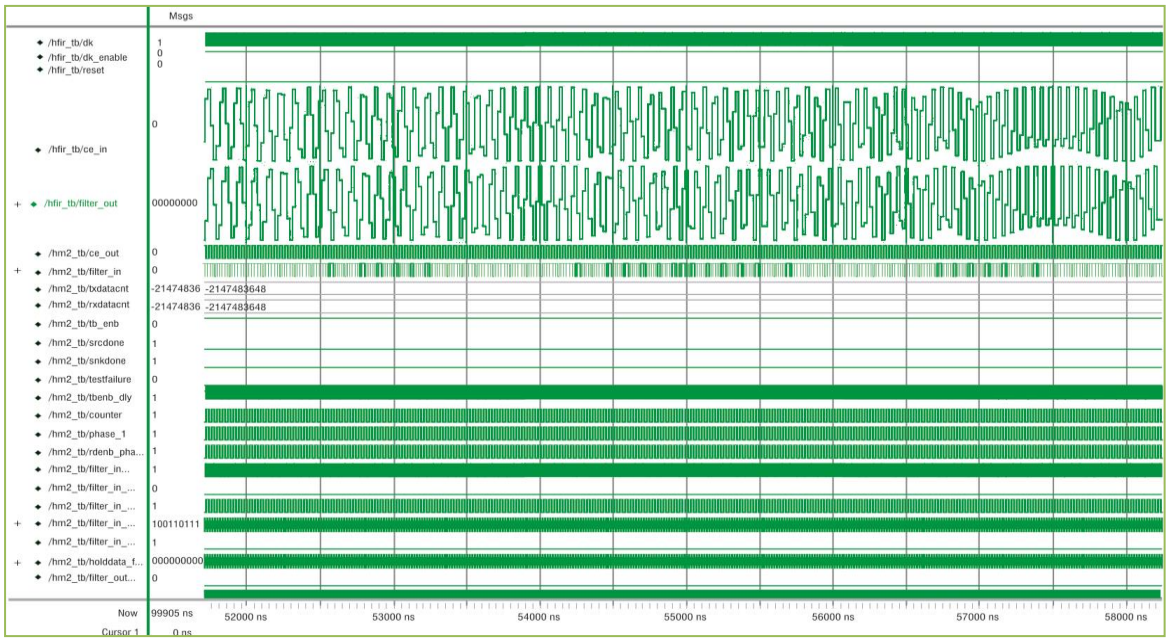

Figure 2.37: Modelsim simulation of Farrow SRC filter design for UMTS

TABLE 2.4
COMPARISON OF IMPLEMENTATION COST AND SPEED ANALYSIS OF FARROW SRC FILTERS

Property	GSM	UMTS
Logic Utilization	<1%	<1%
Worst Case Set up time (t_{su})	5.13ns	4.79ns
Worst Case Clock to Output time (t_{cq})	7.00ns	7.23ns
Clock Frequency	220.80 MHz	306.18 MHz

Thus from the figures 2.34, 2.35, 2.36, 2.37 and table 2.4, it is observed that the implementation of the Farrow filters for the arbitrary sample rate conversion ensures the effective implementation of DUC for the various wireless standards and other applications.

2.14 SUMMARY OF THE CHAPTER

Though, the implementation of signal processing systems on ASICs provides better optimized devices, but the cost of such devices is escalating. Also, the specification alteration requires the complete re-design of the system. With the recent advancements in the FPGA technology, the more complex devices providing the high-speed as required in the DSP applications are at one's disposal. In addition, the FPGA has the advantage of reconfiguration, which provides the upper hand over the ASIC devices. The filter implementation in FPGA, utilizing the dedicated hardware resources can effectively achieve ASIC like performance, while reducing the development time, cost and risks.

In this research work, the multi-stage CIC filters with compensation are implemented in the Altera's Stratix II FPGA for the specifications of the DUC filters of various wireless standards. From these results, it is obvious that the CIC filters reduce the cost of implementation and speed; whereas the compensation techniques ensure the efficient response of the designed filter. Apart from the CIC filters, the Farrow filters are also

implemented efficiently on the FPGA platform for the sample rate conversions. The next chapter deals with the designing of interpolators based on the Farrow filtering structure using Lagrange polynomial. The implementation has been done using the cubic Lagrange as well as the quadratic Lagrange polynomial, and the results have been compared.

CHAPTER 3

DESIGNING OF EFFICIENT FARROW FILTERING STRUCTURE BASED INTERPOLATORS USING LAGRANGE POLYNOMIAL ON FPGA TECHNOLOGY

3.1 INTRODUCTION

The modern trend of higher bandwidth and low power consumption is becoming more and more insistent in the modern digital communication systems. The sample rate conversion plays an important role in the design of advanced communication systems, in which, the basic blocks are the interpolators. The interpolation filters are used at the transmitter side to increase the sampling rate of signal. For fractional rate change, the Farrow structure based interpolation filters are extensively used in the current scenario. But, the ripple contents and computations required in the Farrow filter based interpolators are high. Therefore, there is a need to reduce the ripple contents and computations, which in turn will increase the efficiency and also reduce the cost of communication systems. In this chapter, the efficient designing of Farrow filtering structure based interpolators using the Lagrange polynomial has been proposed on the field programmable gate array (FPGA) platform. There are many factors, which should be kept in mind, while designing on FPGA to make sure that the new devices meet the power and performance necessities of the targeted applications in latest technology. The main factors are selecting the correct technology, choosing the right structural design, applying the accurate software power optimization, enabling easier and power-efficient system design with less computational complexity and resource utilization.

3.2 THE FARROW STRUCTURE

The Farrow structure [25] is an efficient structure to implement the interpolation filter for the fractional rate change as well as for any arbitrary rate change factor. Farrow proposed

a structure of FIR filters in cascade, where the output of each filter is obtained after a delay of single unit from the previous filter output. The Farrow structure is shown in figure 3.1, where $C_0(z), C_1(z), C_2(z), \ldots\ldots\ldots, C_M(z)$ are M+1 FIR filters, and each filter has a poly-phase structure.

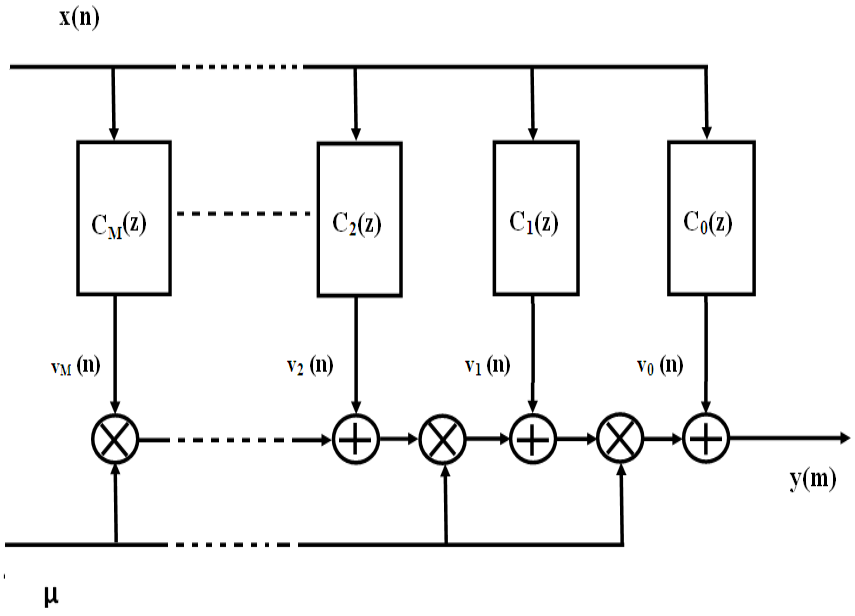

Figure 3.1: The Farrow structure

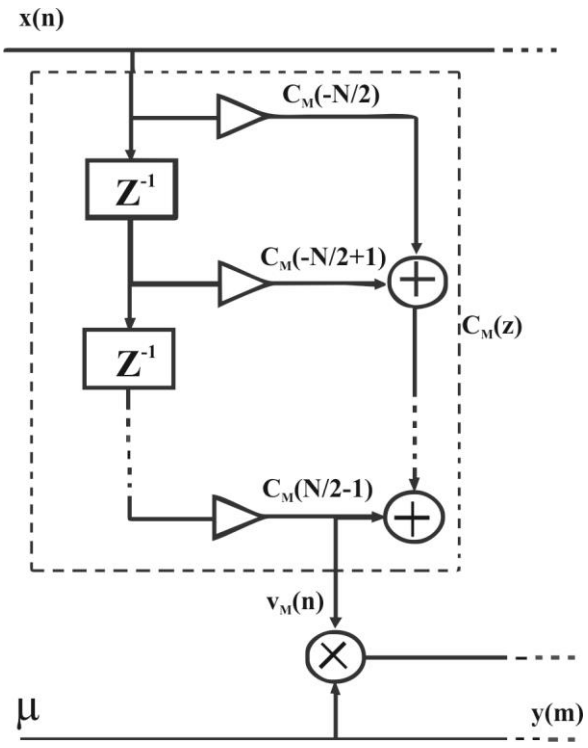

Figure 3.2: Poly-phase structure of M+1th FIR filter

The figure 3.2 shows the poly-phase structure of M+1th FIR filter, where $c_M\left(-\frac{N}{2}\right), c_M\left(-\frac{N}{2}+1\right), \ldots\ldots\ldots c_M\left(\frac{N}{2}-2\right), c_M\left(\frac{N}{2}-1\right)$ are the coefficients of poly-phase structure of M+1th FIR filter, where $v_0(n), v_1(n), v_2(n), \ldots\ldots\ldots, v_m(n)$ are the outputs of FIR filters respectively. The parameter μ is called the fractional interval [131] and has varying values between 0 and 1. It commands the fractional factor by which the input sampled signal is to be interpolated and used to determine the time interval between the interpolated output sample and the previous input sample. The output signal *y(m)* of the Farrow structure relies upon the input sampled signal *x(n)* and response of the FIR filters $h_c(t)$. Mathematically, the output signal *y(m)* is given as

$$y(m) = y_c(t_1) = \sum_{k=-N/2}^{\left(\frac{N}{2}\right)-1} x(n-k)h_c[(k+\mu)T_x] \qquad (3.1)$$

where, it is assumed that *n* is the central sample of the interval $-\frac{NT_x}{2} \le t \le \left(\frac{NT_x}{2}\right) - T_x$ and *N* is an even integer. The impulse response of the Farrow filter is expressed in terms of the coefficients of individual FIR filters, and the factor μ is given as

$$h_c[(k+\mu)T_x] = \sum_{m=0}^{M} c_m(k)\mu^m \qquad for\ k = -\frac{N}{2}, -\frac{N}{2}+1, \ldots\ldots\ldots, \frac{N}{2}-1 \qquad (3.2)$$

where, $c_0(k), c_1(k), \ldots\ldots\ldots, c_M(k)$ are the coefficients and $M \le N-1$ is the degree of polynomial. When the polynomial coefficients are known, it is very easy to compute the impulse response for each interval. We can substitute equation (3.2) in (3.1) to get

$$y(m) = \sum_{k=-N/2}^{\left(\frac{N}{2}\right)-1} x(n-k)[\sum_{m=0}^{M} c_m(k)\mu^m] \qquad (3.3)$$

$$\text{or } y(m) = \sum_{m=0}^{M} \mu^m \left[\sum_{k=-N/2}^{\left(\frac{N}{2}\right)-1} c_m(k)x(n-k)\right] \qquad (3.4)$$

$$\text{or } y(m) = \sum_{m=0}^{M} v_m(n)\mu^m \qquad (3.5)$$

$$\text{where, } v_m(n) = \sum_{k=-N/2}^{\left(\frac{N}{2}\right)-1} c_m(k)x(n-k) \qquad (3.6)$$

and $v_m(n)$ represents the input/output relationship of the each FIR filter shown, whose impulse response coefficients are $c_m(k)$. Hence, the transfer function of each FIR filter is given as

$$C_m(z) = \sum_{k=-\frac{N}{2}}^{\left(\frac{N}{2}\right)-1} c_m(k) z^{-k} \qquad where\ m = 0, 1, 2, \ldots\ldots\ldots, M \tag{3.7}$$

Here, $C_m(z)$ is independent of the value of μ and is fixed for a given design. Now, the only variable parameter in the design is μ. The output signal $y(m)$ is obtained by adding the multiplication of the outputs of FIR filters by the powers of μ respectively.

The Farrow filtering structure based interpolators have been used in the digital audio for transforming the sampling rate from 44.1 kHz to 48 kHz [129]. Other applications of Farrow filtering based interpolators are symbol timing recovery in the QAM demodulation receiver, [73], [141], the echo cancellation in the digital modems, and the sampling rate equalization in WiMax and the GSM communication systems [142]. The generated interpolated signal should possess the same outline as that of the input sampled signal. For this, we have to use some curve fitting or interpolation method with the Farrow filtering structure to approximate the interpolated signal. However, different interpolation methods are applied for the approximation that leads towards the reconstruction of the interpolated signal. The zero order hold interpolation, the linear interpolation [133] and the polynomial interpolation [83], [143] are the different methods of interpolation. The zero order hold as well as linear interpolation are simple methods, and require less computation [133] in comparison to the polynomial interpolation method, but these methods can not provide the precise reconstruction to the interpolated signal. However, the polynomial interpolation method gives accurate shaping to the interpolated signal. Hence in the proposed design, the polynomial interpolation method has been used in Farrow filter [143], [144] to find the new sample values at the arbitrary points between the existing samples. The advantage of doing this is that, it allows the tuning of the

location, at which we wish to interpolate by changing only one parameter i.e., μ in the filter. In the next section, the design of optimum interpolation filter using the Farrow structure has been discussed minutely.

3.3 POLYNOMIAL INTERPOLATION METHOD

Polynomial interpolation method [83], [143] is an efficient method for the interpolation by an arbitrary factor. This procedure is called the polynomial interpolation because it exactly reproduces the given interpolation signal. In this approach, for a given set of the input samples $\{x(-N_1+n), \dots\dots, x(n), \dots\dots, x(n+N_2)\}$, a polynomial approximation or curve fitting $y_c(t)$ to $x_c(t)$ is well defined. Once a polynomial has been fixed, it can be evaluated at any point. Then, the output signal $y(m)$ is determined by sampling $y_c(t)$ at any arbitrary time instant $t = nT_x + \mu T_x$, i.e., the output signal value is calculated mathematically as $y(m) = y_c[(n+\mu)T_x]$. The factor μ is denoted as the fractional interval [144] and $0 \leq \mu \leq 1$. This parameter is used to determine the time interval between the interpolated output sample and the previous input sample. The polynomial approximation $y_c(t)$ to $x_c(t)$ is given as

$$y_c(t) = \sum_{k=-N_1}^{N_2} P(k,t)x(n) \tag{3.8}$$

$$y(m) = y_c[(n+\mu)T_x] \tag{3.9}$$

where, $P(k, t)$ are polynomials and N_1, N_2 are integers. Here, $N=N_2+N_1+1$ is number of consecutive samples surrounding the base point n in such a manner that $-N_1 + n \leq n \leq n + N_2$. The general class of interpolating polynomials those are used for the interpolation are [74], [133], [145]-[147]

a) Lagrange polynomials b) Linear Interpolation

As, the Lagrange polynomials give exact reconstruction of the input samples, they are desirable than the linear interpolation.

3.4 LAGRANGE INTERPOLATION

The Lagrange interpolation [133], [148] method is based on the well-known result in polynomial algebra that uses an N^{th} order polynomial to match N + 1 given arbitrary points. The uniformly spaced Lagrange interpolation formula is defined as

$$P_l(k,t) = \prod_{i=-N_1 \; and \; i \neq k}^{N_2} \frac{t-t_k}{t_k-t_i} \quad for\; k = -N_1, -N_1+1, \dots\dots, N_2-1, N_2 \tag{3.10}$$

where, $P_l(k,t)$ are polynomials in t of order *N-1* or less, which take on the value *1* when $k=t$ and *0* otherwise. This means that the polynomial $P_l(k,t)$ may be expressed by means of the Kronecker delta, i.e.,

$$P_l(k,t) = \partial(k-t) = \begin{cases} 1, & k = t \\ 0, & k \neq t \end{cases}, \; -N_1 \leq k \leq N_2 \tag{3.11}$$

It is easy to write an N^{th} order polynomial [146] that vanishes at the points $k = -N_1, -N_1+1, \dots\dots\dots, N_2-1, N_2$

$$P_l(k,t) = A_n[(t+N_1 \dots\dots (t-N_2+1)(t-N_2)] \tag{3.12}$$

Here, A_n is the scaling constant. Therefore, the requirement of $P_l(k,t)$=1, when t=k is fulfilled only when scaling constant A_n is equal to

$$A_n = \frac{1}{(k+N_1)\dots\dots k(k-1)\dots\dots(k-N_2)} \tag{3.13}$$

If this property is applied to equation (3.8), then it gives

$$y_c(nT_x) = x(n), \qquad where \;\; -N_1 \leq n \leq N_2 \tag{3.14}$$

This depicts that the Lagrange interpolation gives the exact reconstruction of interpolated signal as of the input signal. The Lagrange polynomials are of different orders depending upon the value of *N*. Generally, the lower order polynomials have been preferred, as the higher order polynomials tend to oscillate a lot. In the next section, the Lagrange interpolation method will be applied to the Farrow filter to interpolate the input sampled signal by using an arbitrary factor.

3.5 FARROW FILTERING STRUCTURE FOR LAGRANGE INTERPOLATION

The Lagrange interpolation [145], [147] can be efficiently incorporated using the Farrow filter structures [25]. Even though, the Lagrange interpolation is usually implemented using the direct-form FIR filter structure, where the parameter μ remains constant; but in many applications, it is required to update μ during the run-time. For this, the direct form FIR structure requires more number of multiplication operations, hence consuming more resources. Therefore, it is recommended to implement the Lagrange interpolation using the Farrow filtering structure [74].

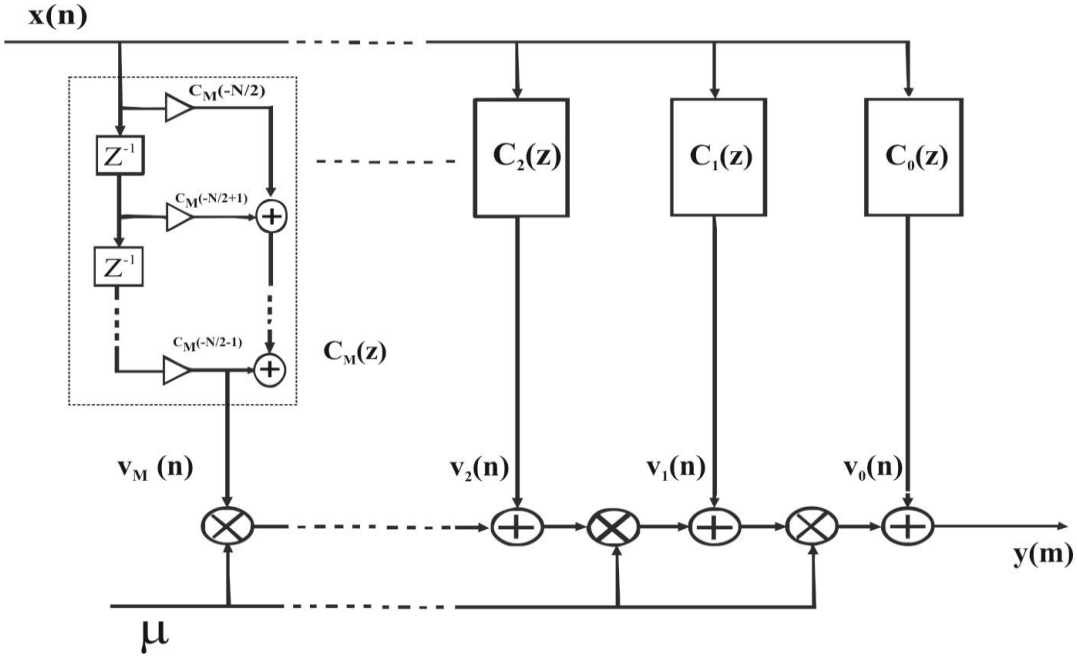

Figure 3.3: The Farrow structure of Lagrange interpolation

Farrow [25] suggested that the FIR interpolating filters using Farrow structure can be expressed as the N^{th} order polynomials in the delay parameter μ, and it is also shown in figure 3.3. Same author also stated that it results in $N + 1$ FIR filters with the constant coefficients [25]. Now, recalling the mathematical expression of the impulse response of Farrow filter having $M+1$ FIR filters from equation (3.2)

$$h_c[(k+\mu)T_x] = \sum_{m=0}^{M} c_m(k)\mu^m \qquad for\ k = -\frac{N}{2}, -\frac{N}{2}+1, \dots\dots\dots, \frac{N}{2}-1 \qquad (3.15)$$

where, $c_0(k), c_1(k), \ldots\ldots\ldots, c_M(k)$ are the coefficients, and $M \leq N-1$ is the degree of polynomial. In *z*-domain, the impulse response is represented as

$$H(z) = \sum_{m=0}^{M} C_m(z)\mu^m \tag{3.16}$$

Using the equation (3.8) and (3.9), the output of Farrow filter is represented in the form of Lagrange polynomial and the input signal [144] as

$$y(m) = y_c(n+\mu) = \sum_{m=0}^{M} P_m(\mu)x(n-m) \tag{3.17}$$

where, $P_m(\mu)$ is the Lagrange polynomial in μ. Equation (3.16) indicates that the higher order polynomials offer better smoothness to the output interpolated signal. The *z*-domain representation of the equation (3.17) is specified as

$$Y(z) = P(\mu)X(z) \tag{3.18}$$

This equation is similar to the output equation of a linear time invariant (LTI) system, whose *z*-domain representation is termed as

$$Y(z) = H(z)X(z) \tag{3.19}$$

By comparing equation (3.17) and (3.18), we can show that

$$P_m(\mu) = \sum_{m=0}^{M} C_m(z)\mu^m \tag{3.20}$$

Equation (3.20) represents a polynomial of degree *M*+1 having the coefficients $C_m(z)$ with unknown values. The solution to this polynomial equation is obtained using the Crammer's rule of matrices. The matrix form of this equation is given as

$$Uc = P \tag{3.21}$$

where, *U* is *LxL* Vandermonde matrix (*L* = *M* +1) set i.e.,

$$U = \begin{bmatrix} 0^0 & 0^1 & 0^2 & \cdots & 0^N \\ 1^0 & 1^1 & 1^2 & \cdots & 1^N \\ 2^0 & 2^1 & 2^2 & \cdots & 2^N \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ N^0 & N^1 & N^2 & \cdots & N^N \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 1 & \cdots & 1 \\ 1 & 2 & 4 & \cdots & 2^N \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 1 & N^1 & N^2 & \cdots & N^N \end{bmatrix} \tag{3.22}$$

and, the coefficient matrix c is represented as

$$c = [C_0(z) \quad C_1(z) \quad \cdots \quad C_N(z)]^T \tag{3.23}$$

The matrix P is known as the transformation matrix, and it is represented as

$$P = \begin{cases} round\ \left[\frac{N}{2}\right]^{n-m} \binom{n}{m}, n \geq m \\ 0, \qquad\qquad\qquad n < m \end{cases} \tag{3.24}$$

The unknown coefficients can be calculated as

$$c = PU^{-1} \tag{3.25}$$

The coefficients of the Farrow filter are then substituted in the impulse response equation and the output interpolated signal is obtained. The Lagrange polynomial of different order can be implemented using the Farrow filtering structure. As second-order or the linear interpolation does not yield exact reconstruction of the interpolated samples, we have to opt the higher order Lagrange interpolation method. In the next section, we will discuss our suggested design for the interpolation filter using the Farrow filtering structure, which is based on the higher order Lagrange interpolation.

3.6 PROPOSED DESIGN AND IMPLEMENTATION OF INTERPOLATION FILTER USING THE FARROW STRUCTURE

In the earlier discussion, it has been described that the Lagrange interpolation is an efficient method to implement the interpolation filter for the arbitrary interpolation factor. Hence, in the proposed design, we will use the Lagrange interpolation method to design the Farrow filter interpolator. The filter will be designed for the third-order or the cubic Lagrange interpolation and the fourth-order or the quadratic Lagrange interpolation method. The Farrow filter coefficients will be calculated by solving the polynomials of third-order and fourth-order.

3.6.1 Design of Farrow Interpolator using Quadratic Lagrange Interpolation Method

The quadratic Lagrange interpolation method is described as the fourth-order Lagrange interpolation. By substituting the order of polynomial as four in the general equation of

the Lagrange interpolation method, we will get the fourth-order Lagrange interpolation. The Farrow structure for this method will be reduced to the fourth-order and having *M+1* i.e., five FIR filters organized in the presented filter structure. The fourth-order Lagrange interpolation method provides better smoothness to the coefficients of Farrow filter, which have been calculated using the matrix equation (3.25) i. e., $c = PU^{-1}$. In this case, the *LxL* Vandermonde matrix will become *5x5* matrix because in this case *M*=4 and *L=M+1*. So, the matrix *U* may be written as

$$U = \begin{bmatrix} 0^0 & 0^1 & 0^2 & 0^3 & 0^4 \\ 1^0 & 1^1 & 1^2 & 1^3 & 1^4 \\ 2^0 & 2^1 & 2^2 & 2^3 & 2^4 \\ 3^0 & 3^1 & 3^2 & 3^3 & 3^4 \\ 4^0 & 4^1 & 4^2 & 4^3 & 4^4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 & 16 \\ 1 & 3 & 9 & 27 & 81 \\ 1 & 4 & 16 & 64 & 256 \end{bmatrix} \tag{3.26}$$

The transformation matrix P is also given as

$$P = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 4 \\ 0 & 0 & 1 & 3 & 5 \\ 0 & 0 & 0 & 1 & 4 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.27}$$

Therefore, the matrix U^{-1} is calculated to be

$$U^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -2.0833 & 4 & -3 & 1.3333 & -0.25 \\ 1.4583 & -4.3333 & 4.75 & -2.3333 & 0.4583 \\ -0.4167 & 1.5 & -2 & 1.1667 & -0.25 \\ 0.0417 & -0.1667 & 0.25 & -0.1667 & 0.0417 \end{bmatrix} \tag{3.28}$$

Hence, the coefficient matrix is given as

$$c = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 4 \\ 0 & 0 & 1 & 3 & 5 \\ 0 & 0 & 0 & 1 & 4 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -2.0833 & 4 & -3 & 1.3333 & -0.25 \\ 1.4583 & -4.3333 & 4.75 & -2.3333 & 0.4583 \\ -0.4167 & 1.5 & -2 & 1.1667 & -0.25 \\ 0.0417 & -0.1667 & 0.25 & -0.1667 & 0.0417 \end{bmatrix} \tag{3.29}$$

$$c = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -0.25 & -0.8334 & 1.5 & -0.5 & 0.0834 \\ 0.4167 & -0.6668 & 0 & 0.3333 & -0.0832 \\ -0.2499 & 0.8332 & -1 & 0.4999 & -0.0832 \\ 0.0417 & -0.1667 & 0.25 & -0.1667 & 0.0417 \end{bmatrix} \tag{3.30}$$

These are optimum coefficients of the FIR filter stages [74] in the Farrow structure calculated in our proposed design, using the fourth-order Lagrange polynomial. These coefficients will be fed into the corresponding FIR filter stages to calculate the output interpolated signal and the frequency response of the Farrow interpolator. The proposed Farrow interpolator has been designed using the Altera DSP builder advanced block set in MATLAB, which is a nano-meter based FPGA device. The design set-up used to implement the fourth-order Lagrange interpolation has been shown in figure 3.4.

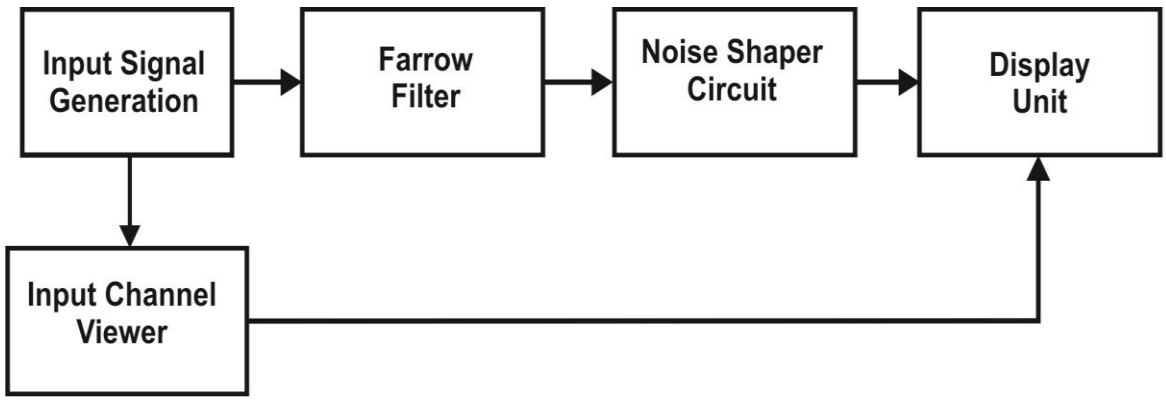

Figure 3.4: Interpolation filter design using Farrow structure

The input signal to the Farrow filter is a single frequency sine wave, with the sampling frequency set at 1 MHz. The parameter μ has been set to 0.111 and it can also be modified in between the run-time. The output interpolated signal is of 1.111 MHz frequency. The Farrow structure based polynomial FIR filters operate at the input sample rate, while the parameter μ ($0 \leq \mu \leq 1$) operates at the output sample rate. The number of cycles between the two unique data samples is different for the polynomial FIR filters and the time offset (μ) generation. Therefore, we can not directly combine the two data paths in the Farrow filtering structure. Before the data paths are combined, we must synchronize the two signals. The valid signal affirmed in the input signal generation provides synchronization.

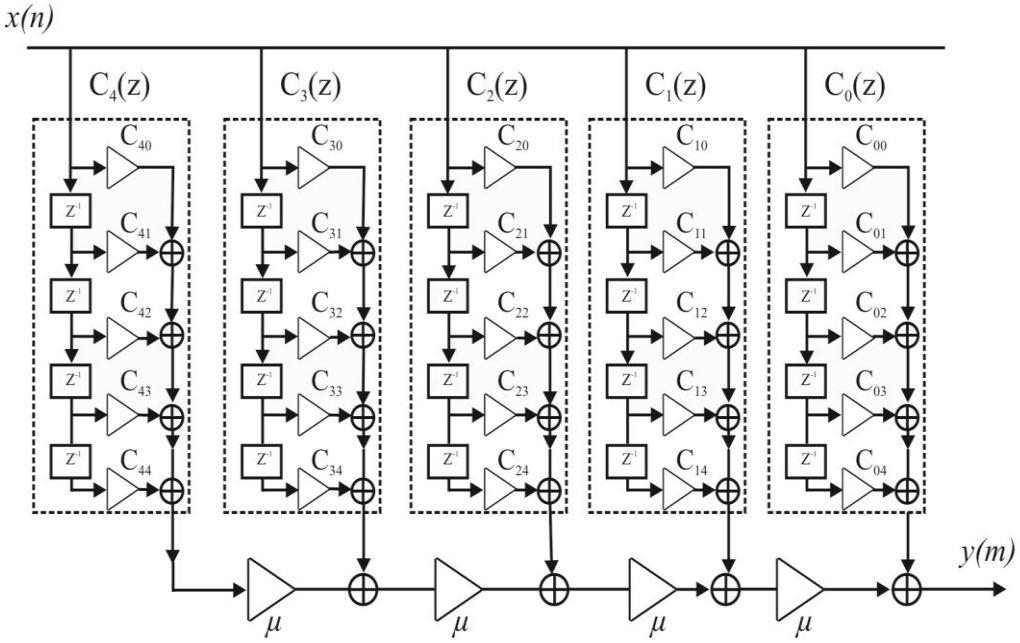

Figure 3.5: Farrow structure for quadratic Lagrange interpolation

The optimal Farrow filter coefficients calculated in the equation (3.30) have been substituted in the corresponding FIR filter structures. The inside view of the Farrow filter in set-up has been plotted in figure 3.5, while the mapping of the filter coefficients with the corresponding FIR filter has been shown in the table 3.1.

TABLE 3.1
MAPPING OF QUADRATIC LAGRANGE COEFFICIENTS WITH FIR FILTER STAGES

	C(0)	*C(1)*	*C(2)*	*C(3)*	*C(4)*
$h_4(n)$	0.0417	-0.1667	0.25	-0.1667	0.0417
$h_3(n)$	-0.0832	0.4999	-1	0.8332	-0.2499
$h_2(n)$	-0.0832	0.3333	0	-0.6668	0.4167
$h_1(n)$	0.0834	-0.5	1.5	-0.8334	-0.25
$h_0(n)$	0	0	0	1	0

The overall response has been calculated at the output of interpolation filter. The frequency response of the Farrow interpolation filter has been shown in the figure 3.6. The response depicts that the output spectrum of filter has an out of band power, when compared to the input response. According to the design standards, the output spectrum

should follow the input spectrum, and it should not have out of band power because it will cause the loss of signal power. Hence, our motive is not only just to interpolate the signal, but also to take care that the output spectrum should follow the input spectrum.

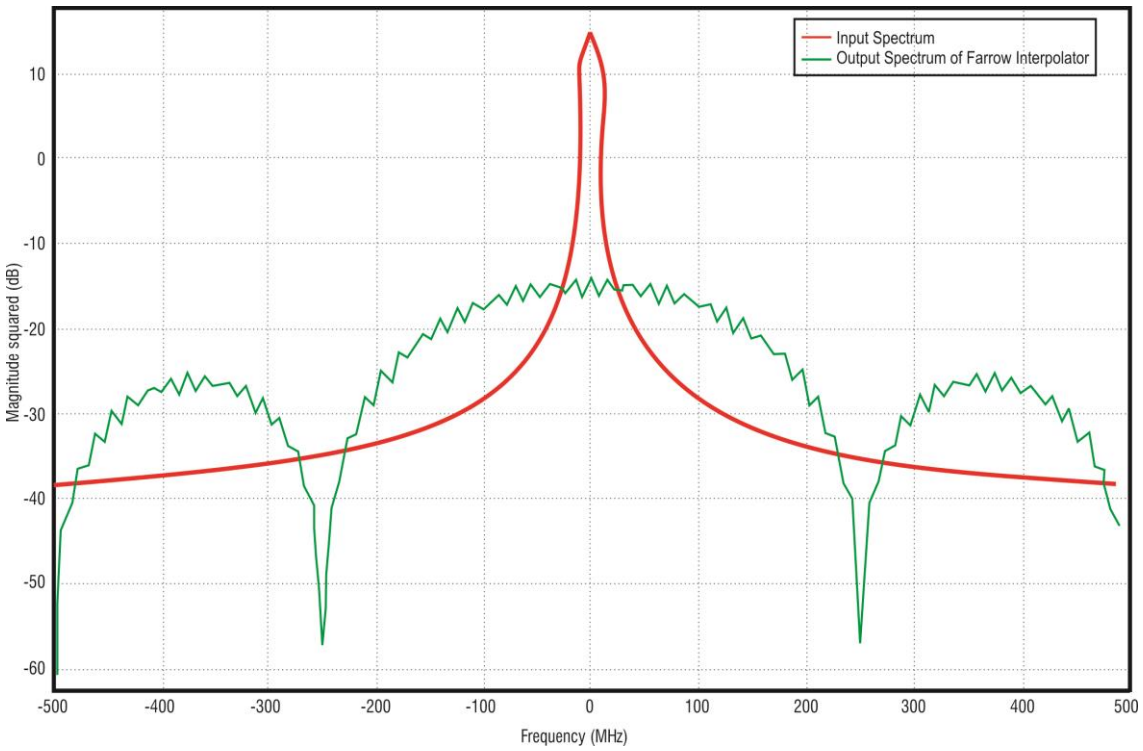

Figure 3.6: Frequency response of the Farrow interpolation filter for quadratic Lagrange interpolation

To mitigate the effects of out of band power of the filter output, we have used a noise shaper at the output of Farrow filter in the design set-up (shown in figure 3.4). The noise shaper block [149] is used to suppress the noise present in the Farrow filter output. This block consists of a channel viewer and a root raised cosine (RRC) filter [131]. The RRC filter helps to reduce the noise level in the data signal. The channel viewer de-serializes the bus on its input to produce a configurable number of output signals, those are not using the time division multiplexing (TDM). When a single channel input is present, the channel viewer block strips out all the invalid samples. The invalid samples are those samples, which are not synchronized with the output sampling rate. The input and output signals of the noise shaper block are presented in the table 3.2.

TABLE 3.2
PORT INTERFACE FOR THE NOISE SHAPER BLOCK

Signal	Direction	Description
q	Input	The data input to the block.
v	Input	Indicates validity of data input signals. If v is high then the data on the wire is valid.
c	Input	Indicates channel of data input signals. If v is high, then c indicates from which channel the data corresponds to.
c_n	Output	Each output is a noise suppressed and de-serialized version of the data signal contained in the TDM bus. The output value is updated on each clock cycle that has valid data when the channel matches the required channel.

The overall response of the Farrow interpolator with the noise shaper has been shown in figure 3.7. It has been observed that the output spectrum follows the input signal spectrum.

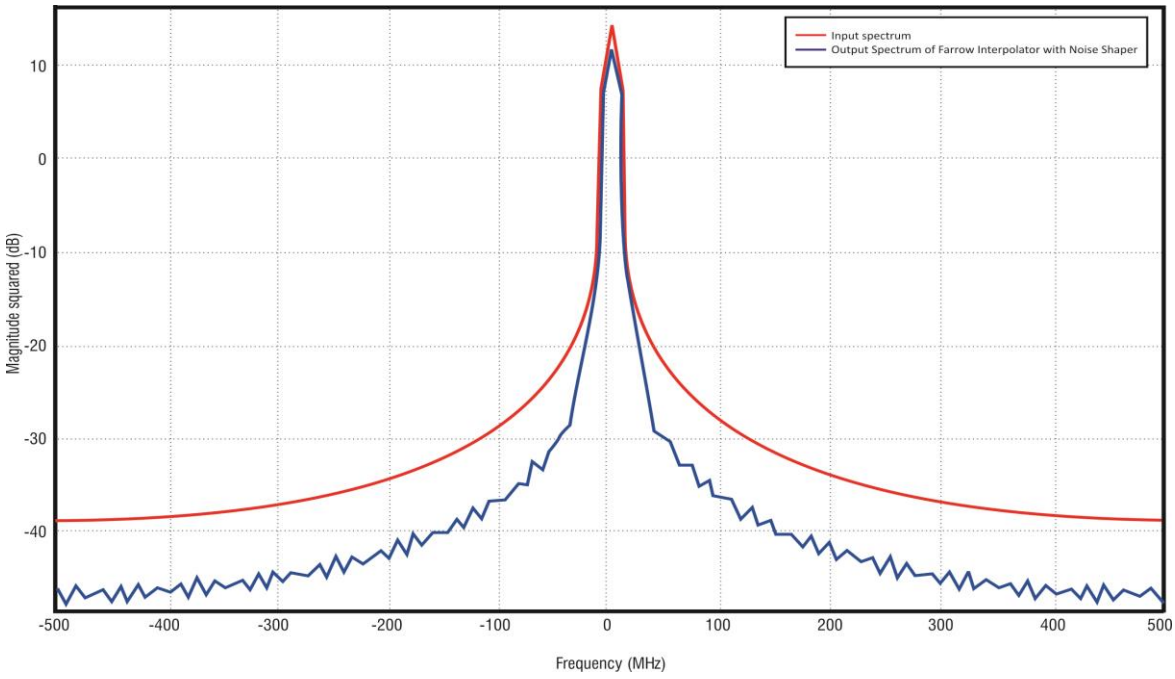

Figure 3.7: Overall Farrow interpolation filter response with noise shaper for quadratic Lagrange interpolation

Hence through the introduction of the noise shaper, the overall filter response has been improved a lot. In the stop-band of output spectrum, the ripple level is more as compared to the input spectrum. The ripple level has been calculated as approximately 60 dB. Because of the introduction of noise shaper, the frequency response of the filter follows the input signal spectrum. The comparison of the Farrow filter response with and without noise shaper is demonstrated in figure 3.8.

This comparison graph indicates that the spectrum of Farrow interpolator with noise shaper results in the optimum design features. But there is some ripple content in the stop-band, which can be reduced by using other Lagrange interpolation techniques. Therefore, the design of Farrow interpolator using the quadratic Lagrange interpolation coefficients is very efficient in terms of the frequency spectrum characteristics.

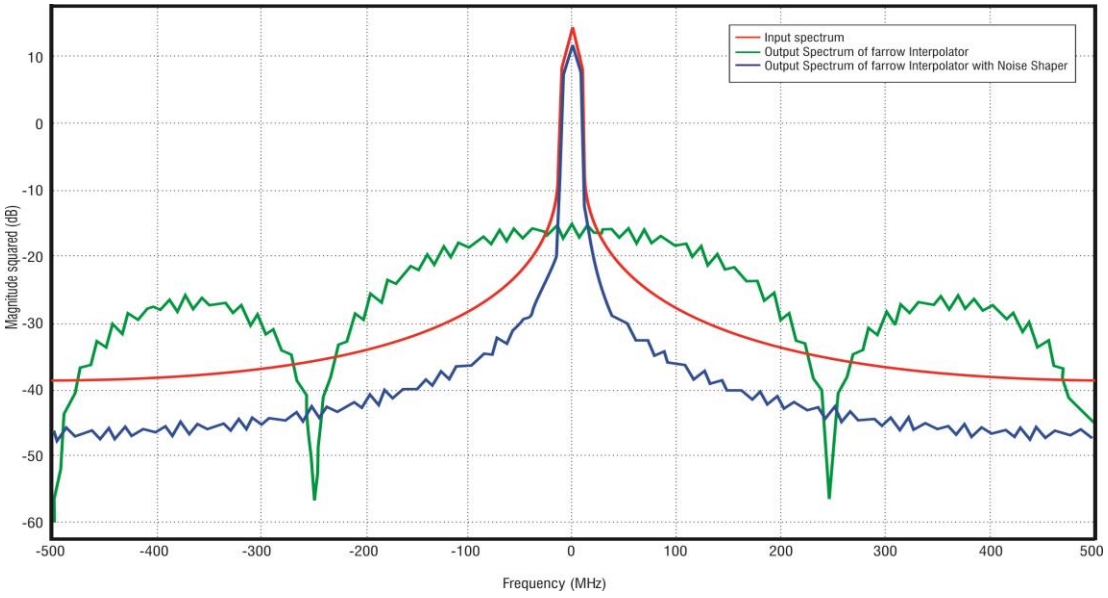

Figure 3.8: Comparison of output spectrum of Farrow interpolator for quadratic Lagrange interpolation

The computational requirements of the Farrow interpolator for the quadratic Lagrange interpolation are 19 multiplications and 16 additions per output sample. In the next segment, we will use the cubic Lagrange interpolation method to design the Farrow filter interpolator.

3.6.2 Design of the Farrow Interpolator using Cubic Lagrange Interpolation Method

The cubic Lagrange interpolation method is described as the third-order Lagrange interpolation. By substituting the order of polynomial as three in the general equation of the Lagrange interpolation method, we will get the third-order Lagrange interpolation. The Farrow structure for this method will be reduced to third-order, and it will have *M*+1 i.e., four FIR filters arranged in the resultant filter structure. The coefficients of the Farrow filter have been calculated again by using the matrix equation (3.25) i. e., $c = PU^{-1}$. In this case, the *LxL* Vandermonde matrix becomes a 4x4 matrix because *M*=3 and *L*=*M*+1. Therefore, the matrix *U* may be given as

$$U = \begin{bmatrix} 0^0 & 0^1 & 0^2 & 0^3 \\ 1^0 & 1^1 & 1^2 & 1^3 \\ 2^0 & 2^1 & 2^2 & 2^3 \\ 3^0 & 3^1 & 3^2 & 3^3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 \\ 1 & 3 & 9 & 27 \end{bmatrix} \tag{3.31}$$

The transformation matrix *P* is also formulated as

$$P = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.32}$$

and, the matrix U^{-1} is calculated to be

$$U^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1/6 & 3 & -3/2 & 1/3 \\ 1 & -5/2 & 2 & -1/2 \\ -1/6 & 1/2 & -1/2 & 1/6 \end{bmatrix} \tag{3.33}$$

Therefore, the coefficient matrix is stated as

$$c = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1/6 & 3 & -3/2 & 1/3 \\ 1 & -5/2 & 2 & -1/2 \\ -1/6 & 1/2 & -1/2 & 1/6 \end{bmatrix} \tag{3.34}$$

$$c = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1/3 & -1/2 & 1 & -1/6 \\ 1/2 & -1 & 1/2 & 0 \\ -1/6 & 1/2 & -1/2 & 1/6 \end{bmatrix} \tag{3.35}$$

These are the optimum coefficients of FIR filter stages in the Farrow structure calculated for our proposed design using the third-order Lagrange polynomial. These coefficients are fed into the corresponding FIR filter stages to calculate the output interpolated signal and also the frequency response of Farrow interpolator. The proposed Farrow interpolator design set-up for the third-order Lagrange interpolation is also similar as shown in the figure 3.7. The only difference that exists between the two designs is the number of FIR filter stages. In this case, the number of FIR filter stages is reduced to four as shown in figure 3.9. The inside view of Farrow filter in the set-up has also been shown in the figure 3.9; where $C_0(z), C_1(z), C_2(z)$ and $C_3(z)$ are the FIR filters with corresponding coefficients. The mapping of filter coefficients with the corresponding FIR filter has been shown in the table 3.3.

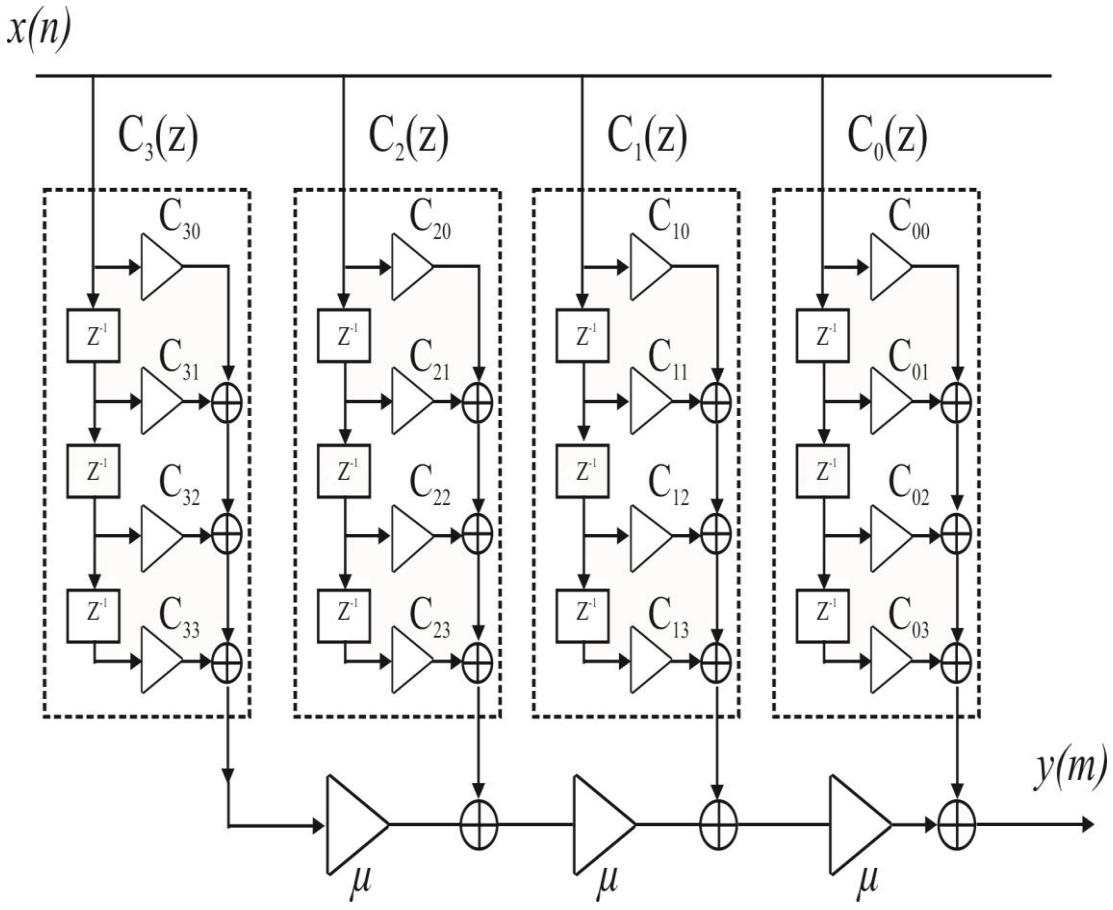

Figure 3.9: Farrow structure for cubic Lagrange interpolation

TABLE 3.3

MAPPING OF CUBIC LAGRANGE COEFFICIENTS WITH FIR FILTER STAGES

	C(0)	*C(1)*	*C(2)*	*C(3)*
$h_3(n)$	1/6	-1/2	1/2	-1/6
$h_2(n)$	0	1/2	-1	1/2
$h_1(n)$	-1/6	1	-1/2	-1/3
$h_0(n)$	0	0	1	0

The complete response has been calculated at the output of interpolation filter. The frequency response of the Farrow interpolation filter has been showcased in figure 3.10. The response manifests that the output spectrum of Farrow filter has out of band power, when compared to the input signal spectrum.

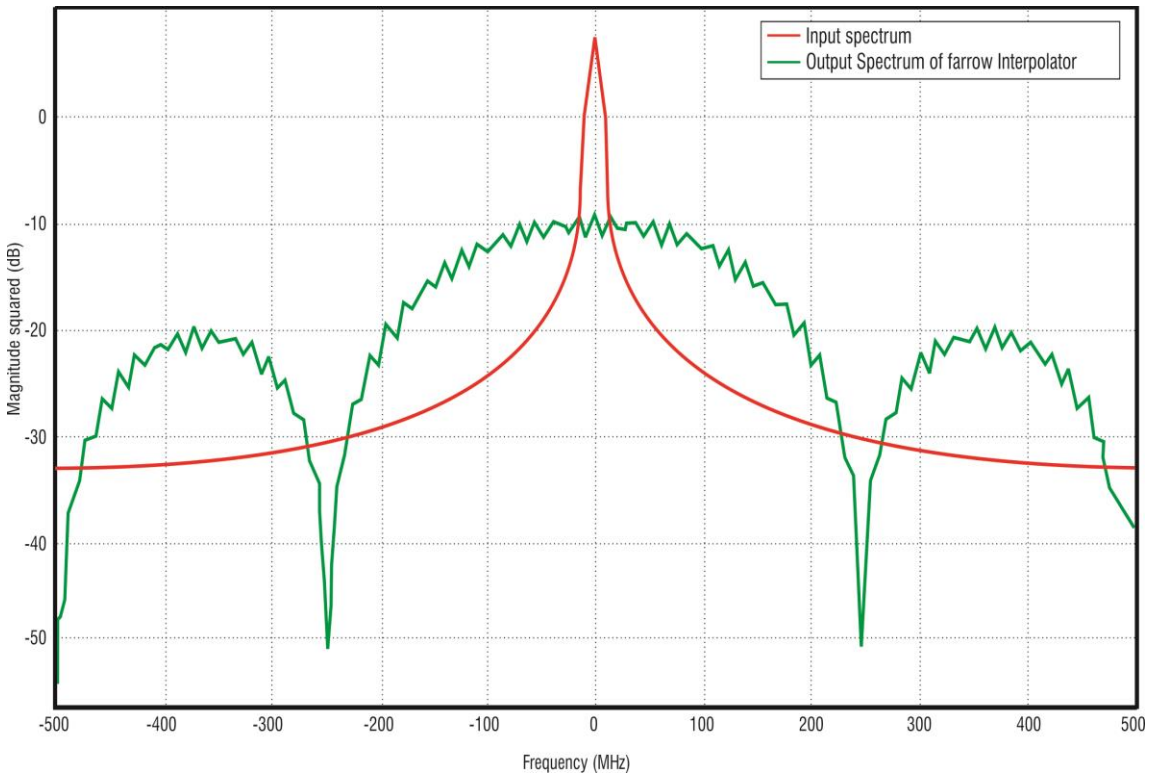

Figure 3.10: Frequency response of the Farrow interpolation filter for cubic Lagrange interpolation

Therefore, we have to add the noise shaper in set-up, so that the output spectrum follows the input signal spectrum. The spectrum of Farrow interpolator with the noise shaper is plotted in figure 3.11. It has been found that the spectrum for the cubic Lagrange

interpolation is better than that of the quadratic Lagrange interpolation because the ripple content in this case is suppressed to approximately 20-30 dB. This is because the higher order polynomials tend to oscillate more as compared to the lower order polynomials.

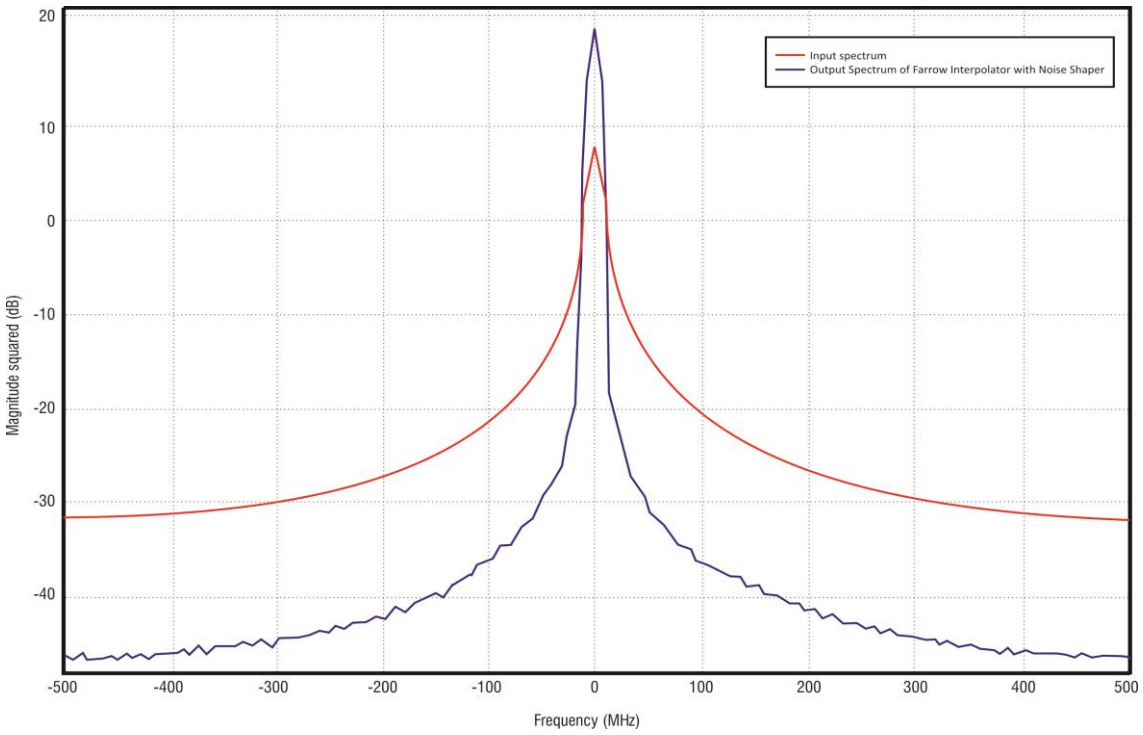

Figure 3.11: Overall Farrow interpolation filter response with noise shaper for cubic Lagrange interpolation

Hence, the cubic Lagrange coefficients are considered as optimum coefficients for the proposed design. With the introduction of noise shaper, the response of the filter follows the input signal spectrum. The comparison of the Farrow filter response with and without noise shaper is expressed in figure 3.12. The comparison graph indicates that the spectrum of Farrow interpolator with noise shaper results in the optimum design characteristics. In this case, the ripple level in the stop-band is very less as compared to the quadratic Lagrange interpolation spectrum. The computational burden of the Farrow interpolator for cubic Lagrange interpolation is 11 multiplications and 9 additions per output sample. The computational requirements in this case are also less as compared to the previous case.

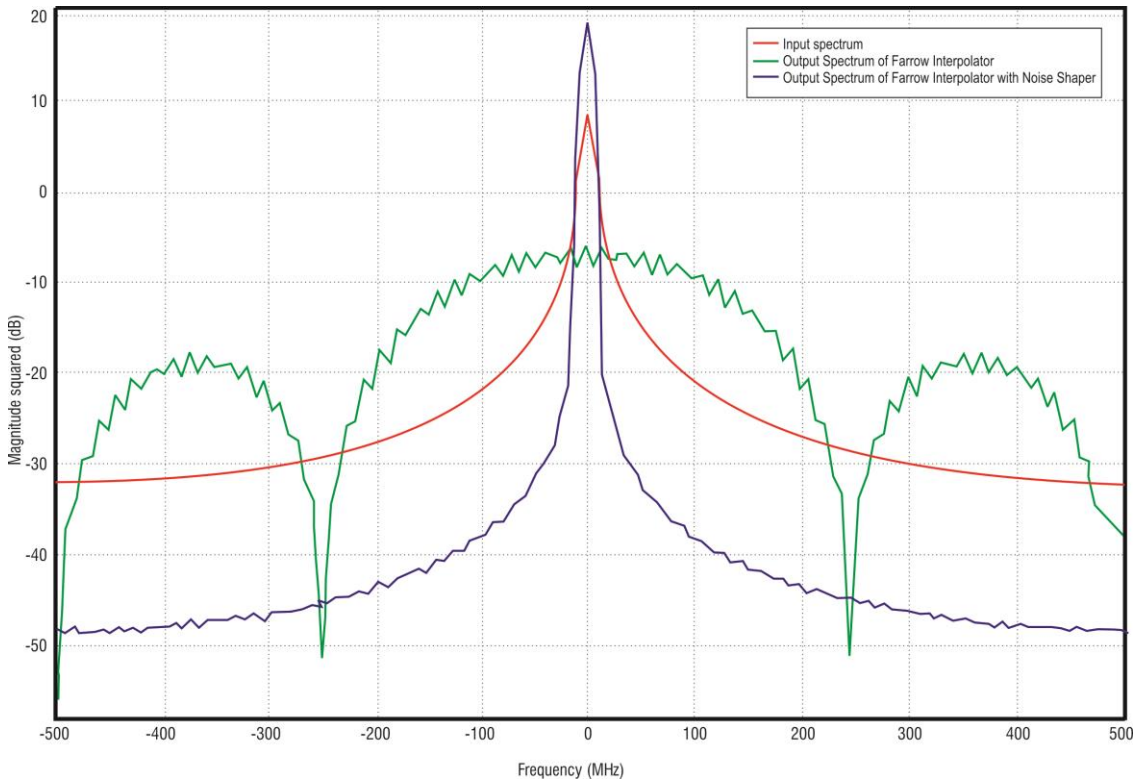

Figure 3.12: Comparison of output spectrum of Farrow interpolator for cubic Lagrange interpolation

In the next section, the comparison of two techniques has been deliberated to find out the optimal solution for the design of Farrow filter interpolator.

3.7 COMPARISON OF THE QUADRATIC LAGRANGE INTERPOLATION AND CUBIC LAGRANGE INTERPOLATION DESIGN TECHNIQUES FOR FARROW STRUCTURE BASED INTERPOLATORS

The proposed Farrow interpolator has been designed using the quadratic Lagrange interpolation and the cubic Lagrange interpolation. Using these two techniques, the optimal Farrow filter coefficients have been calculated, and are fed to the corresponding filter stages of the Farrow filter. The response curves of both the methods have been attained, and it has been found that the cubic Lagrange interpolation method is the most efficient method for the design of optimum Farrow filter interpolator. The comparison chart between these two techniques has been shown in the table 3.4.

TABLE 3.4
COMPARISON OF THE PROPOSED INTERPOLATION TECHNIQUES

Comparison parameter	Quadratic Lagrange	Cubic Lagrange
No. of FIR filter stages of Farrow filter	5	4
Coefficient matrix	5x5	4x4
Reconstruction of the Interpolated signal	Provides better smoothness	Somewhat less compared to Quadratic
Computational Needs	19 multiplications 16 additions	11 multiplications 9 additions
Ripple level	60 dB	20-30 dB

The above comparison indicates that even though the cubic Lagrange interpolation method provides somewhat less smooth interpolated signal as compared to the quadratic interpolation method, but the ripple factor, computational needs and the resource utilization of the cubic Lagrange interpolation method are less. Hence, the Farrow interpolation filter for the cubic Lagrange interpolation is the best method for the design of interpolation filter using Farrow structures.

3.8 DESIGN AND IMPLEMENTATION OF MODIFIED FARROW STRUCTURE

In the previous section, it has been shown that for the arbitrary fractional rate conversion, the cubic Lagrange polynomial is the finest method for realization of the interpolation filters. Therefore, a technique has been projected to design the efficient Farrow interpolators with the help of cubic Lagrange interpolation using the modified Farrow structures. Let us consider the $L \times L$ matrix U, such that

$$U = \begin{bmatrix} 0^0 & 0^1 & 0^2 & \cdots & 0^N \\ 1^0 & 1^1 & 1^2 & \cdots & 1^N \\ 2^0 & 2^1 & 2^2 & \cdots & 2^N \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ N^0 & N^1 & N^2 & \cdots & N^N \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 1 & \cdots & 1 \\ 1 & 2 & 4 & \cdots & 2^N \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 1 & N^1 & N^2 & \cdots & N^N \end{bmatrix} \tag{3.36}$$

For *N*=*3*, the matrix *U* can be shown as

$$U = \begin{bmatrix} 0^0 & 0^1 & 0^2 & 0^3 \\ 1^0 & 1^1 & 1^2 & 1^3 \\ 2^0 & 2^1 & 2^2 & 2^3 \\ 3^0 & 3^1 & 3^2 & 3^3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 \\ 1 & 3 & 9 & 27 \end{bmatrix} \tag{3.37}$$

The co-efficient matrix can be represented as

$$c = [C_0(z) \quad C_1(z) \quad \cdots \quad C_N(z)]^T \tag{3.38}$$

The coefficient matrix c can be calculated by $c = U^{-1}z$. However, the U^{-1} is calculated as

$$U^{-1} = \begin{bmatrix} 1.0000 & 0 & 0 & 0 \\ -1.8333 & 3.0000 & -1.5000 & 0.3333 \\ 1.0000 & -2.5000 & 2.0000 & -0.5000 \\ -0.1667 & 0.5000 & -0.5000 & 0.1667 \end{bmatrix} \tag{3.39}$$

Therefore, the coefficient matrix may be calculated by

$$\begin{bmatrix} c_0(z) \\ c_1(z) \\ c_2(z) \\ c_3(z) \end{bmatrix} = \begin{bmatrix} 1.0000 & 0 & 0 & 0 \\ -1.8333 & 3.0000 & -1.5000 & 0.3333 \\ 1.0000 & -2.5000 & 2.0000 & -0.5000 \\ -0.1667 & 0.5000 & -0.5000 & 0.1667 \end{bmatrix} \begin{bmatrix} 1 \\ z^{-1} \\ z^{-2} \\ z^{-3} \end{bmatrix} \tag{3.40}$$

Thus, we get *c(z)* as

$$\begin{bmatrix} c_0(z) \\ c_1(z) \\ c_2(z) \\ c_3(z) \end{bmatrix} = \begin{bmatrix} 1 \\ -1.8333 + 3z^{-1} - 1.5\, z^{-2} + 0.3333z^{-3} \\ 1 - 2.5\, z^{-1} + 2\, z^{-2} - 0.5\, z^{-3} \\ -0.1667 + 0.5\, z^{-1} - 0.5\, z^{-2} + \ 0.1667z^{-3} \end{bmatrix} \begin{bmatrix} 1 \\ z^{-1} \\ z^{-2} \\ z^{-3} \end{bmatrix} \tag{3.41}$$

The total transfer function can be written as

$$H(z) = c_0(z) + c_1(z)\mu + c_2(z)\mu^2 + c_3(z)\mu^3 \tag{3.42}$$

The Farrow structure designed from equation (3.42) can be made more efficient by altering the range of parameter μ. The new parameter range is $0 \leq \ \mu \leq \ 1$ (for odd *N*) or $-0.5 \leq \ \mu \leq \ 0.5$ (for even *N*).

As, we are considering *N*=3, therefore the value of μ will lie between 0 and 1. In any practical communication system, the value of μ may vary between 0 and 1 for the arbitrary fractional rate change. An arbitrary value of μ has been considered as 0.9286,

and for this particular value of μ, the Farrow filtering structure proposed in [25] and the modified Farrow structure presented in [40] have been designed and compared for the cubic Lagrange polynomial interpolation on the FPGA platform. Thus, by keeping μ as 0.9286, the equation (3.42) may be rewritten as

$$H(z) = 1 + 0.9286(-1.83333 + 3z^{-1} - 1.5z^{-2} + 0.3333z^{-3}) + (0.9286)^2(1 - 2.5z^{-1} + 2z^{-2} - 0.5z^{-3}) + (0.9286)^3(-0.1667 + 0.5z^{-1} + 0.5z^{-2} + 0.1667z^{-3} \quad (3.43)$$

By considering the value of μ as 0.9286, the original (unmodified) Farrow structure for the cubic Lagrange polynomial interpolation may be designed as shown in figure 3.13.

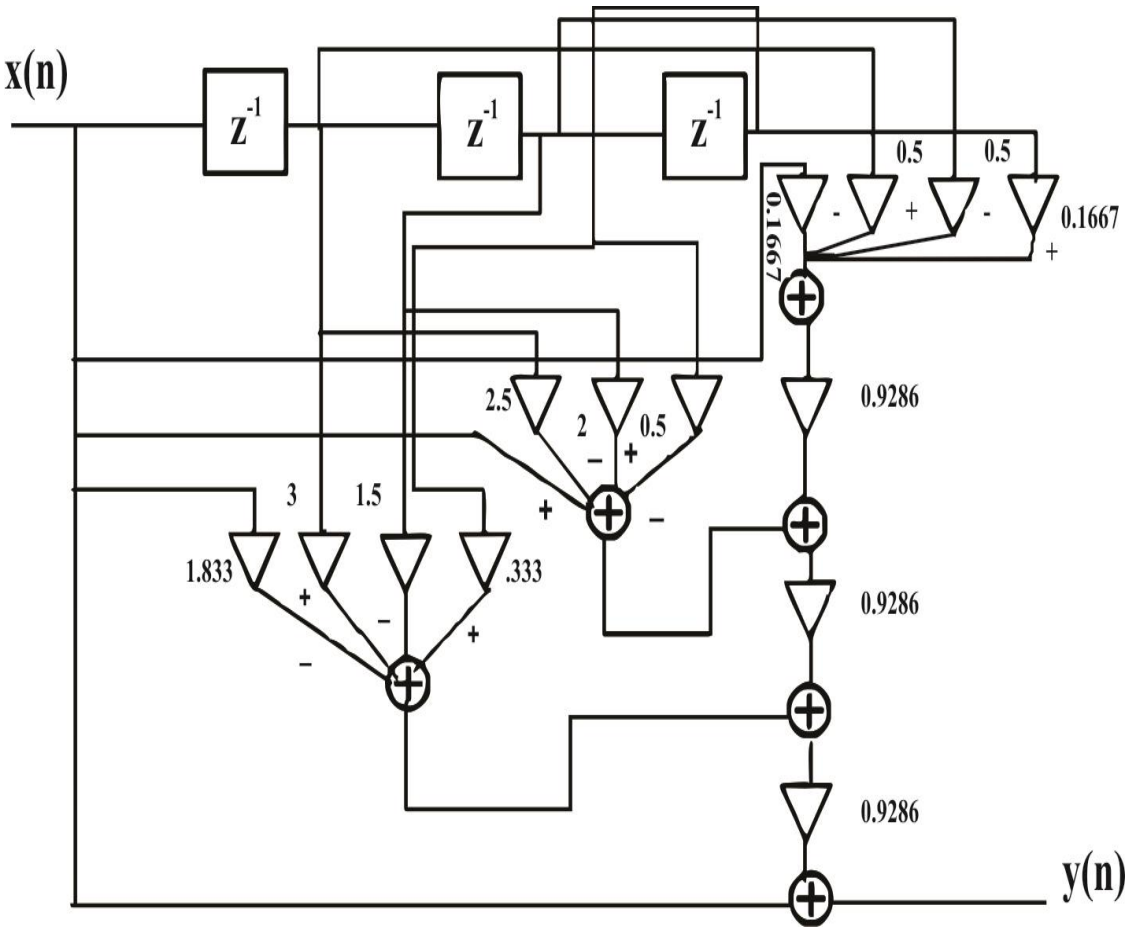

Figure 3.13: Original (unmodified) Farrow structure for third-order Lagrange polynomial

By using the same value of μ as 0.9286, the modified Farrow structure for the cubic Lagrange polynomial interpolation may be designed as shown in figure 3.14.

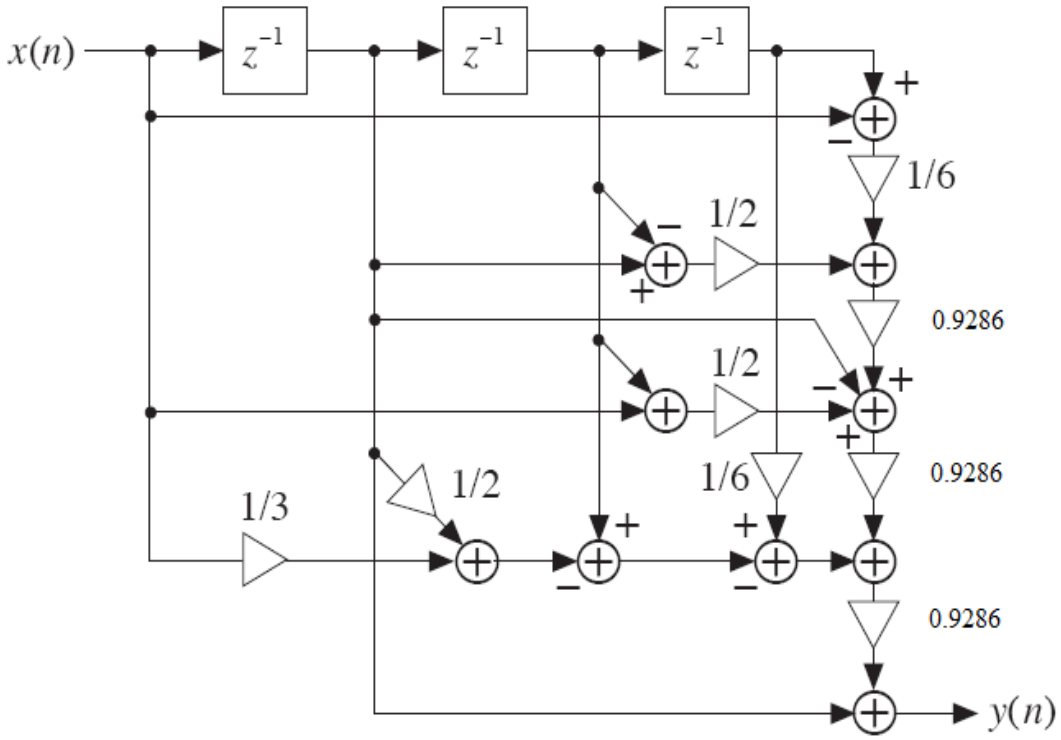

Figure 3.14: Modified Farrow structure for third-order Lagrange polynomial

The modified Farrow structure requires only 10 additions and 9 multiplications as compared to the 6 additions and 14 multiplications required by the unmodified Farrow structure (demonstrated in the figure 3.13). Therefore, we may observe that the computational complexity reduces drastically by using the cubic Lagrange polynomial interpolation. The spectrum of the input and output of original (unmodified) Farrow filtering structure for the cubic Lagrange polynomial interpolation is shown in the figure 3.15, and the spectrum of the input and output of modified Farrow structure for the cubic Lagrange polynomial interpolation is shown in figure 3.16.

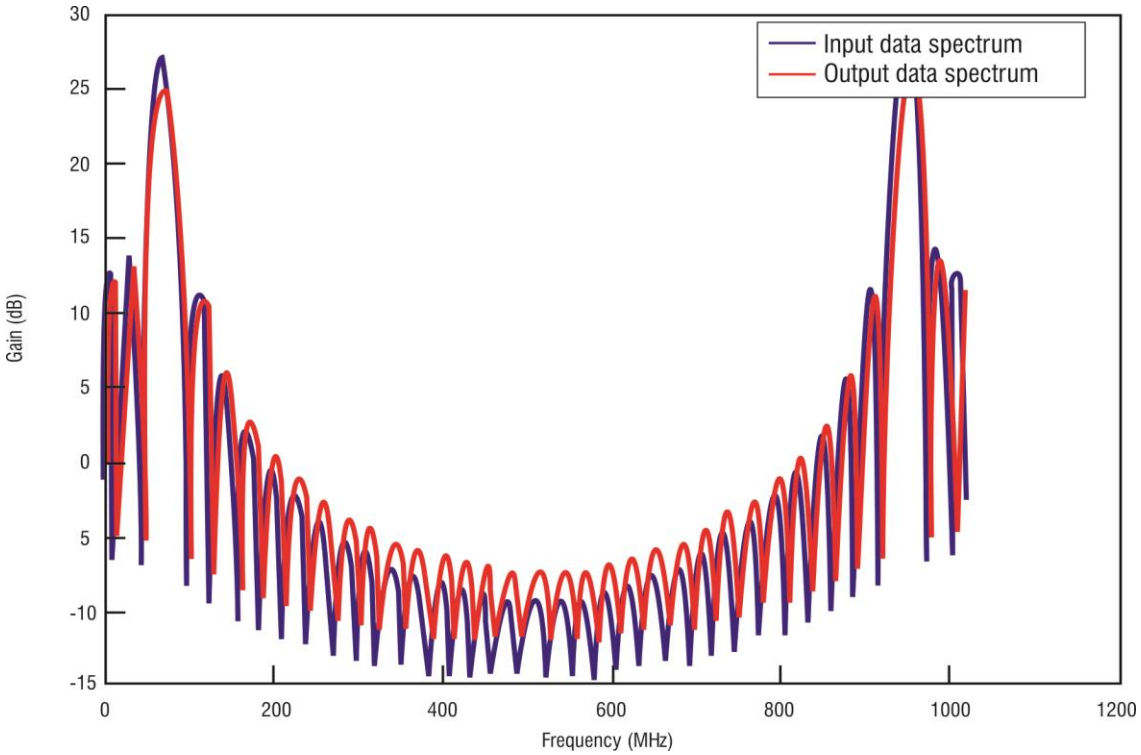

Figure 3.15: Spectrum of the input and output of original (unmodified) Farrow structure for third-order Lagrange polynomial

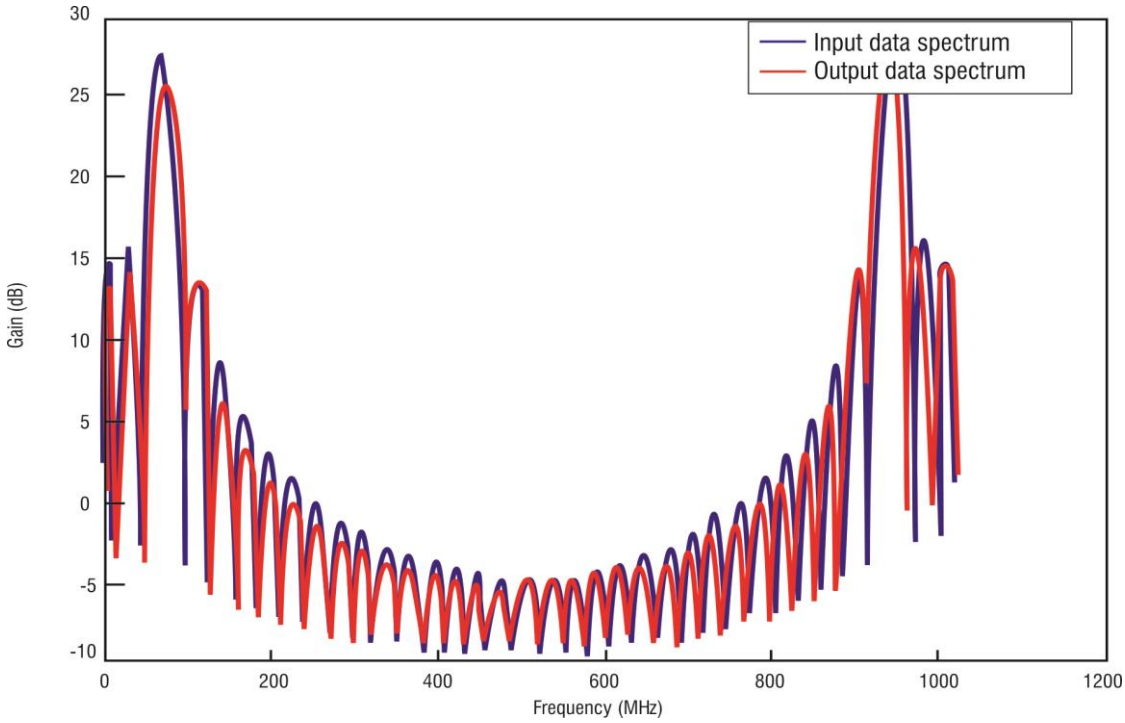

Figure 3.16: Spectrum of the input and output of modified Farrow structure for third-order Lagrange polynomial

For the same value of μ (i.e. 0.9286), the significant difference between the gains in figure 3.15 and figure 3.16 can be noticed. The minimum value of gain for the output spectrum improves from -12.2 dB in case of the original (unmodified) Farrow structure to -9.1 dB in case of the modified Farrow structures for the cubic Lagrange polynomial interpolation. Also, the figure 3.16 depicts the superiority of the proposed modified structure in terms of the spectral re-growth. The modified Farrow structure for the cubic Lagrange polynomial interpolation has also been implemented on the Altera Stratix III EP3SL70F780C2N device for calculating the resource utilization. The table 3.5 demonstrates the comparison of resources utilized by the modified and the unmodified Farrow structures.

TABLE 3.5
COMPARISON OF RESOURCES UTILIZED BY THE UNMODIFIED AND PROPOSED MODIFIED FARROW STRUCTURES

Design Elements	**LUTs**		**Multipliers**		**Memory Bits**	
	Without Modification	**Proposed Structure**	**Without Modification**	**Proposed Structure**	**Without Modification**	**Proposed Structure**
Farrow Filter	16768	16300	64	59	1728	1689
Input channel Viewer	119	89	---	---	---	---
Noise shaper	124	120	---	---	---	---
Total	17011	16509	64	59	1728	1689

From the results presented in table 3.5, it can be concluded that the number of look-up tables (LUTs) for the proposed design decreases by approximately 2.95%, the number of multipliers decreases by approximately 7.81% and the memory bits decrease by approximately 2.25%. Hence, for its implementation, the presented design requires relatively less FPGA area.

3.9 SUMMARY OF THE CHAPTER

In this research work, the design of optimum interpolation filter using the Farrow filter structures has been presented. The interpolation filter has been designed using the Lagrange interpolation method, because it gives the exact shaping to the interpolated signal as of the input signal. The two Lagrange polynomial interpolation methods namely the cubic Lagrange interpolation and the quadratic Lagrange polynomial have been conferred. The Farrow interpolation filters have been designed using the optimal filter coefficients calculated by the cubic as well as the quadratic polynomial interpolation method. The comparison of these design methods has also been discussed, which shows that even though the cubic Lagrange interpolation method provides somewhat less smooth interpolated signal than the quadratic Lagrange interpolation, but the spectrum of interpolated signal has very low ripple content when compared to the quadratic Lagrange interpolation spectrum. Also, the computational requirements of the cubic Lagrange interpolation method are less as compared to the quadratic Lagrange interpolation. Therefore, the cubic interpolation method is found to be the optimum solution for the design of Farrow interpolation filter.

Further, the design and implementation of interpolation filter with the help of the modified Farrow structure has been implemented using the third-order Lagrange polynomial. The performance of proposed interpolation filter using the modified Farrow structure has been compared with an interpolation filter, in which the unmodified Farrow structure is being used. The proposed structure outperforms the unmodified Farrow structure in terms of the spectral re-growth and less area requirements, when implemented on the FPGA platform. In the next chapter, the efficient implementation of digital interpolation systems for the up-sampling of QAM/QPSK based multi-stage digital interpolators has been presented.

CHAPTER 4

PERFORMANCE COMPARISON OF QUADRATURE AMPLITUDE MODULATION (QAM)/ QUADRATURE PHASE SHIFT KEYING (QPSK) BASED SINGLE AND DOUBLE-STAGE DIGITAL INTERPOLATORS

4.1 INTRODUCTION

Harmonization of system parameters at the transmitting and receiving ends of a channel is noteworthy to set up the consistent communication. The mismatches in the carrier frequency, carrier phase or timing errors in the signal sampling can considerably degrade communication. In this chapter, an efficient approach has been presented for the multi-stage interpolation structures, those can be used in the digital receivers for the improvement of noise performance and to reduce the symbol synchronization errors. The process of digital signal interpolation is a fundamental technique in the field of signal processing. It is used in many contexts, most typically for the conversion between sampling rates.

The digital up-converter (DUC) provides interpolation, filtering, frequency translation and summing of DUC channel outputs to produce the digital intermediate frequency (IF) outputs. The DUCs can be found in the wireless base-station transmit electronics, specialized digital IF receive-test equipment, cable television modulators and software defined radio transmission equipments. The number of DUC channels is based on the number of frequency carriers, the number of phased or diverse antennas and antenna delay calibration outputs. The test equipments and digital radio electronics typically use DUC devices combined with the analog to digital converter (ADC) and digital to analog converter (DAC) to provide the digital radio sections.

4.2 MULTI-STAGE IMPLEMENTATION OF SAMPLING RATE CONVERSION

In practical applications of sampling rate conversion, we frequently come across the decimation factors and interpolation factors, which are much larger than unity. For instance, assume that there is an assignment of varying the sampling rate by the factor L/M=130/63.

where, L is interpolation factor and M is the decimation factor. Although in theory, this rate can be achieved precisely, the implementation would necessitate a bank of 130 poly-phase filters, and it may be computationally inefficient. In this section, the emphasis is on the consideration of methods for performing the sampling rate conversion for either $M>>1$ and/or $L>>1$ in the multiple stages [141]. Initially, consider the interpolation by a factor $L>>1$, and let us assume that L can be factored into a product of the positive integer as

$$L = \prod_{i=1}^{J} L_i \tag{4.1}$$

Then, the interpolation by a factor L can be obtained by cascading J stages of interpolation and filtering, as shown in figure 4.1.

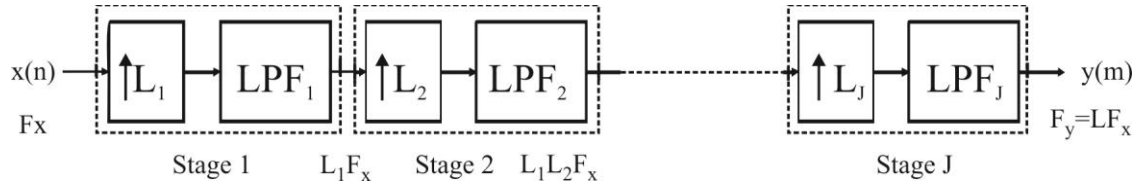

Figure 4.1: Multi-stage implementation of interpolation by a factor *L*

It can be seen that the filter in each of the interpolators roots out the images produced by the up-sampling process in the corresponding interpolator. In a comparable manner, the decimation by a factor M, where M may be factored into a product of positive integer as

$$M = \prod_{i=1}^{K} M_i \tag{4.2}$$

and therefore decimation by a factor *M* can be accomplished as a cascade of *K* stages of filtering and decimation as illustrated in figure 4.2.

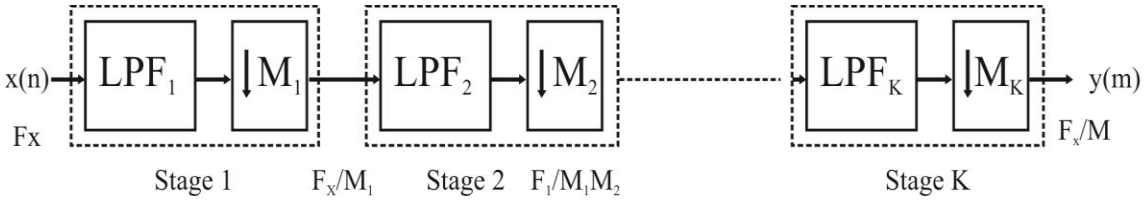

Figure 4.2: Multi-stage implementation of decimation by a factor M

Thus, the sampling rate at the output of i^{th} stage is

$$F_i = \frac{F_{i-1}}{D_i}, \qquad i = 1, 2, 3 \ldots\ldots\ldots\ldots\ldots\ldots K \tag{4.3}$$

where, the input rate for the sequence $\{x(n)\}$ is $F_O = F_X$. The obvious question that arises from the above discussion is that why we are considering such multi-stage structures. At first glance, it would appear as if, we are greatly increasing the overall computation (as we have incorporated filters between the each stage) of the structure. However, this is precisely the opposite of what occurs in practice. The reasons for considering multi-stage structures, of the types shown in figure 4.1 and figure 4.2 are to reduce the computations while implementing the underlying system, reduced storage in the system, simplified filter design problem (i.e., wider normalized transition-band allowed at each stage), and the reduced finite-word-length effects (i.e., round-off noise and coefficient sensitivity) in the implementations of digital filters [150].

4.3 FILTER REQUIREMENTS FOR MULTI-STAGE DESIGNS

There are a large number of parameters and trade-offs involved in the design of multi-stage decimation and interpolation systems, including the choice of number of stages, the decimation or interpolation ratios M_J or L_K at each stage, the filter requirements and the actual filter designs at each stage. Several general design procedures have been advised for choosing these parameters by Crochiere and Rabiner in [106]. In this section, we define the terminology used to specify the filter requirements for each stage of a multi-

stage system. To ensure that no aliasing occurs in the overall decimation process, we can design each filter stage to avoid aliasing within the frequency band of interest. To elaborate, let us define the pass-band and the transition-band in the overall decimator as

Pass-band : $0 \leq F \leq F_P$

Transition-band :$F_P \leq F \leq F_S$

where, (stop-band edge) $F_S \leq F_X/2M$ and F_P is the pass-band edge. Then, aliasing in the band $0 \leq F \leq F_S$ is evaded by selecting the frequency bands of each filter stage as follows

Pass-band : $0 \leq F \leq F_P$

Transition-band : $F_P \leq F \leq F_i - F_S$

Stop-band : $F_i - F_S \leq F \leq \frac{F_{i-1}}{2}$

For example, in the first filter stage $F_1 = F_x/M_1$, and the filter is designed to have the following frequency bands

Pass-band :$0 \leq F \leq F_P$

Transition-band : $F_P \leq F \leq F_1 - F_S$

Stop-band : $F_1 - F_S \leq F \leq \frac{F_0}{2}$

After decimation by M_1, there is aliasing from the signal components that falls in the filter transition-band, but the aliasing occurs at the frequencies above F_S. Thus, there is no aliasing in the frequency band $0 \leq F \leq F_S$. By designing the filters in the subsequent stages to satisfy the specifications given above, it is ensured that no aliasing occurs in the primary frequency band $0 \leq F \leq F_S$. By decimating in the multiple stages, the width of transition region can be augmented through a reduction in the sampling rate. In the case of a multi-stage interpolator, the sampling rate at the output of i^{th} stage is

$$F_{i-1} = L_i F_i for i = L, L-1, \ldots\ldots\ldots\ldots\ldots\ldots 1 \quad (4.4)$$

and, the output rate is

$$F_O = LF_J \tag{4.5}$$

when the input sampling rate is F_J, the corresponding frequency band specifications are

$$\text{Pass-band} \quad : \ 0 \leq F \leq F_P$$

$$\text{Transition-band} \quad : \ F_P \leq F \leq F_i - F_S$$

By interpolation at the multiple stages, we are able to decrease the width of the transition region through an increment in the sampling rate.

4.4 MULTI-STAGE NARROW-BAND FILTERS

This method is convenient for the low-pass, high-pass and band-pass filters having the bandwidths lower than one fourth of the sampling rate. Those filters are considered as narrow-band filters. Let us examine the multi-stage filtering approach in the case of a low-pass filter with a low cut-off frequency [130], [151]-[152]. A general structure of the multi-stage multi-rate filter, consisting of a decimator, kernel filter $H_K(z)$, and the interpolator is shown in figure 4.3.

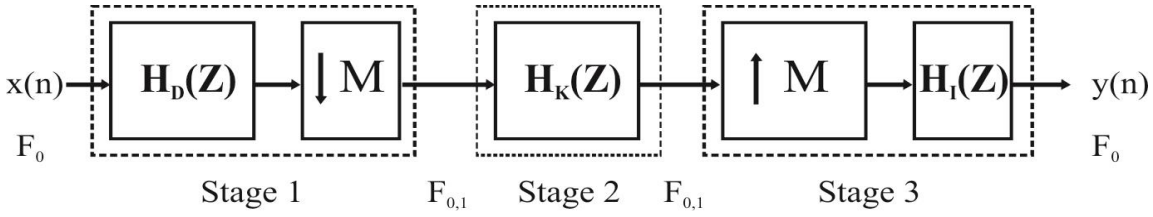

Figure 4.3: Multi-stage filter: decimator, kernel filter, and interpolator

The sampling rate F_0 of the input signal $\{x[n]\}$ is first reduced to a lower rate $F_{0,1} = F_0/M$. The actual filtering with the kernel filter $H_K(z)$ is performed at this lower rate, and the original rate F_0 is reinstated by interpolation. The transfer function of the overall multi-rate filter of the figure 4.3 is given as

$$H(z) = \frac{Y(z)}{X(z)} \tag{4.6}$$

which is composed of three transfer functions $H_D(z)$, $H_K(z)$ and $H_I(z)$. Under the assumption that $H_D(z)$ and $H_I(z)$ eliminate the aliasing effects, the overall transfer function $H(z)$ is expressed as

$$H(z) = H_D(z)H_K(z^M)H_I(z) \tag{4.7}$$

and accordingly, with the replacement $z = e^{j\omega}$, we obtain the frequency response of the multi-rate filter as

$$H\left(e^{j\omega}\right) = H_D(e^{j\omega})H_K(e^{jM\omega})H_I(e^{j\omega}) \tag{4.8}$$

The figure 4.4 indicates the magnitude responses of filters from figure 4.3. The pass-band and stop-band edge frequencies of the kernel filter F_P and F_S respectively, are identical to those of the desired overall filter. The frequency F_P is also the pass-band edge of the decimation filter $H_D(z)$ and of the interpolation filter $H_I(z)$. For a fixed conversion factor M, which determines the lower sampling rate $F_{0,1} = F_0/M$, the stop-band edge frequency for $H_D(z)$ and $H_I(z)$ is located at $F_{0,1}–F_S$, as indicated in figure 4.4.

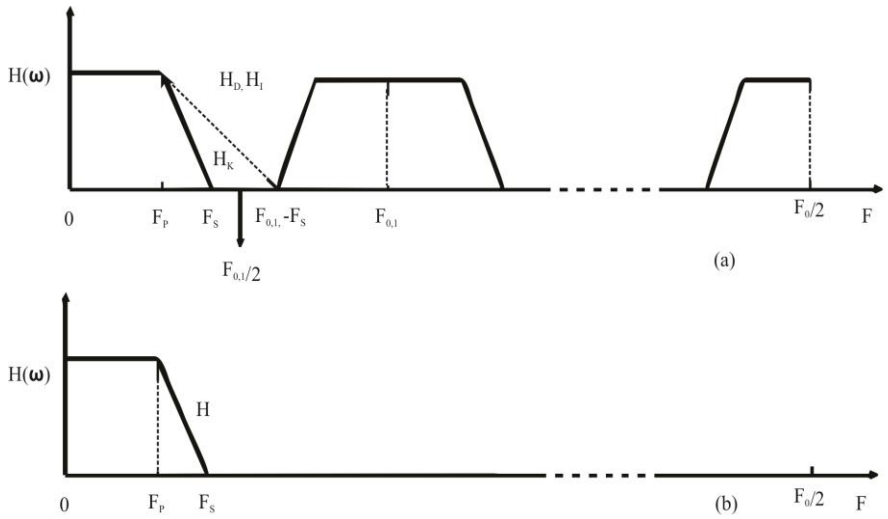

Figure 4.4: (a) Magnitude responses of decimation and interpolation filters $H_D(z)$ and $H_I(z)$, and of the kernel filter $H_K(z)$.

(b) Magnitude response of the overall filter H(z).

The figure 4.4 illustrates that a chain consisting of a decimation filter, kernel filter and interpolation filter achieves a narrow-band characteristics due to the usage of three filters of significantly relaxed specifications. The overall magnitude response is the product of the magnitude responses $|H_K(e^{jM\omega})|$, $|H_D(e^{j\omega})|$ and $|H_I(e^{j\omega})|$. The minimum stop-band attenuation of the overall filter is determined by the stop-band attenuations of $H_K(z^M)$,

$H_D(z)$ and $H_I(z)$. In the range $Fs \leq F \leq F_{0,1}–Fs$, the minimum stop-band attenuation of the overall filter is determined by the attenuation of the kernel filter $H_K(z^M)$. For frequencies above $F_{0,1}–F_S$, the minimum stop-band attenuation is determined by the cascade of decimation and interpolation filters $H_D(z)$ and $H_I(z)$. It is noteworthy that the stop-band attenuation of $H_D(z)$ and $H_I(z)$ is not only to ensure the stop-band of the overall filter significantly, but also the purpose of $H_D(z)$ is to suppress aliasing in decimation and $H_I(z)$ to remove the imaging in interpolation.

4.5 INTERPOLATORS IMPLEMENTATION

The interpolation of a signal by an integer up-sample factor can be accomplished by processing the signal $x(n)$, with the cascade of an expander and low-pass filter, as shown in figure 4.5. If the input signal $x(n)$ operates on the sampling frequency f, this achieves interpolated output signal $y(n)$ at the increased sampling frequency L_f. More intricate interpolation systems can be designed as the cascade of multiple expanders and low-pass filters on a system containing two expanders and two low-pass filters, as shown in figure 4.6.

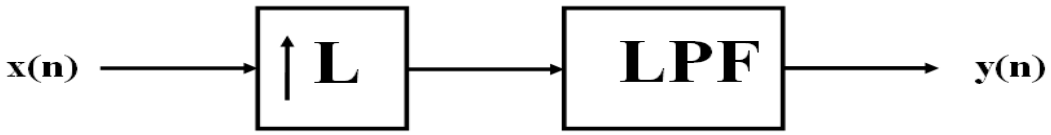

Figure 4.5: Single-stage interpolation of the signal *x(n)*

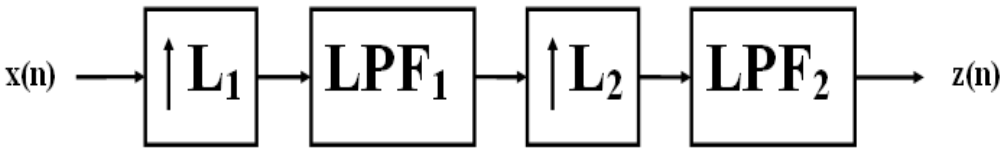

Figure 4.6: Two-stage interpolation of the signal *x(n)*

If the parameters of cascaded interpolator in figure 4.6 are chosen correctly, such that $L1.L2 = L$ and with suitable choices of the low pass filters LPF_1 and LPF_2, then this system will achieve equivalent interpolation to the system in figure 4.5 [134], [150]. In

this case, assuming the input sampling frequency *f*, the interpolated output signals *y(n)* and *z(n)* both have sampling frequencies L_f, and more specifically *y(n) = z(n)*. Therefore, these two systems are distinct designs attaining the same interpolation and can be compared in terms of the computational efficiency [153]. The figure 4.7 shows a block diagram of the digital IF subsystem with DUC.

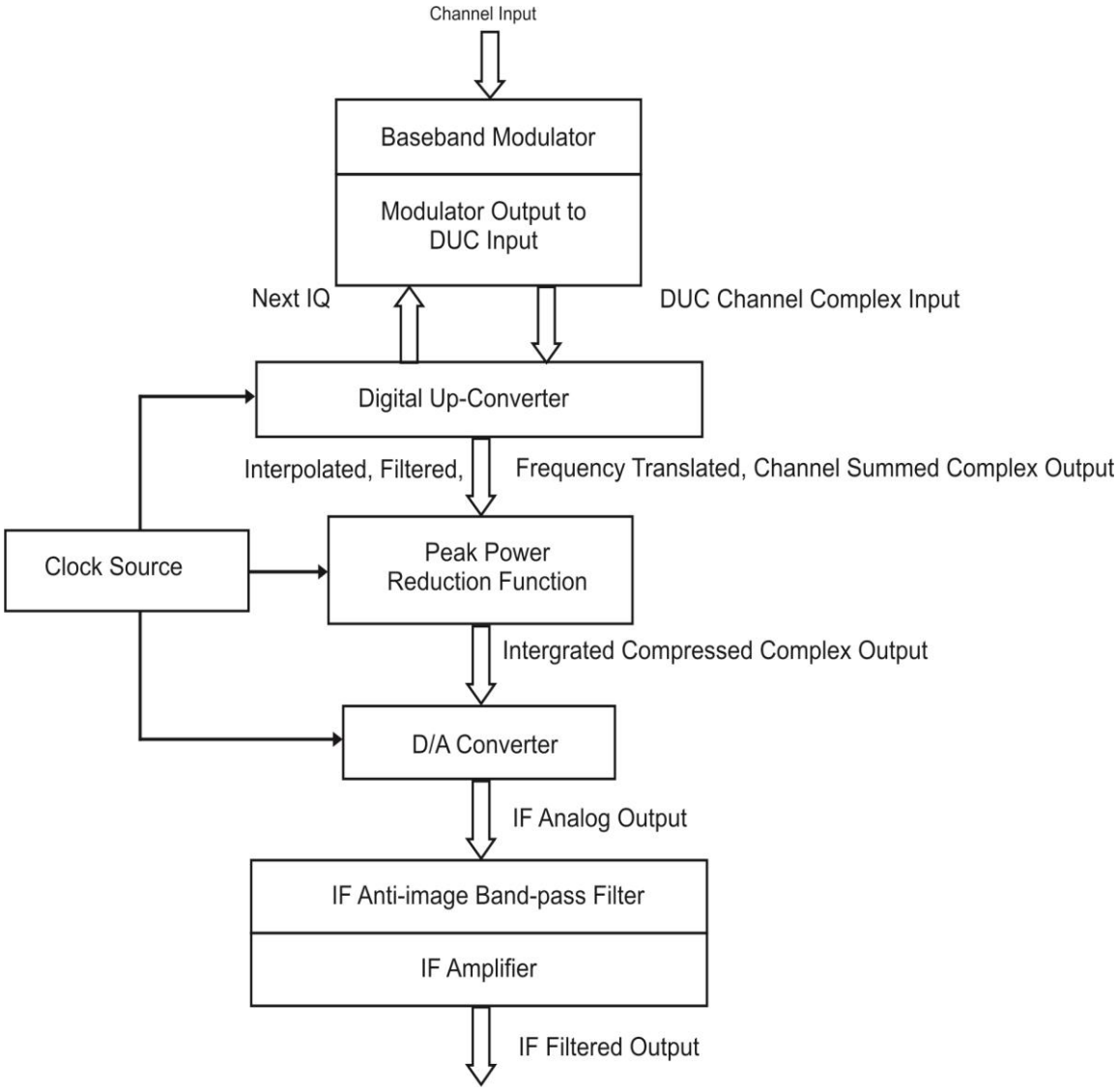

Figure 4.7: Digital IF subsystem with DUC

The baseband modulator converts the channel signal into a complex baseband set of in-phase/quadrature (IQ) signals. The modulated output to the DUC input block buffers the modulator output. The next IQ signal synchronizes the DUC request for data, with the customised logic. The DUC general functions are to interpolate, filter and translate the

baseband digital data, and it also adds the selected DUC channels for a real or complex output. The DUC output is sent to the peak power reduction function and/or the DAC [14], [112], [113]. The DAC offers further interpolation, filtering, mixing and complexity to get analog IF output by generating the real conversion process. The high quality clock source provides sample timing to the DAC and a digital clock to the DUC [12], [36]. The analog IF output from the DAC is filtered and amplified. The following figure 4.8 depicts the two-stage DUC.

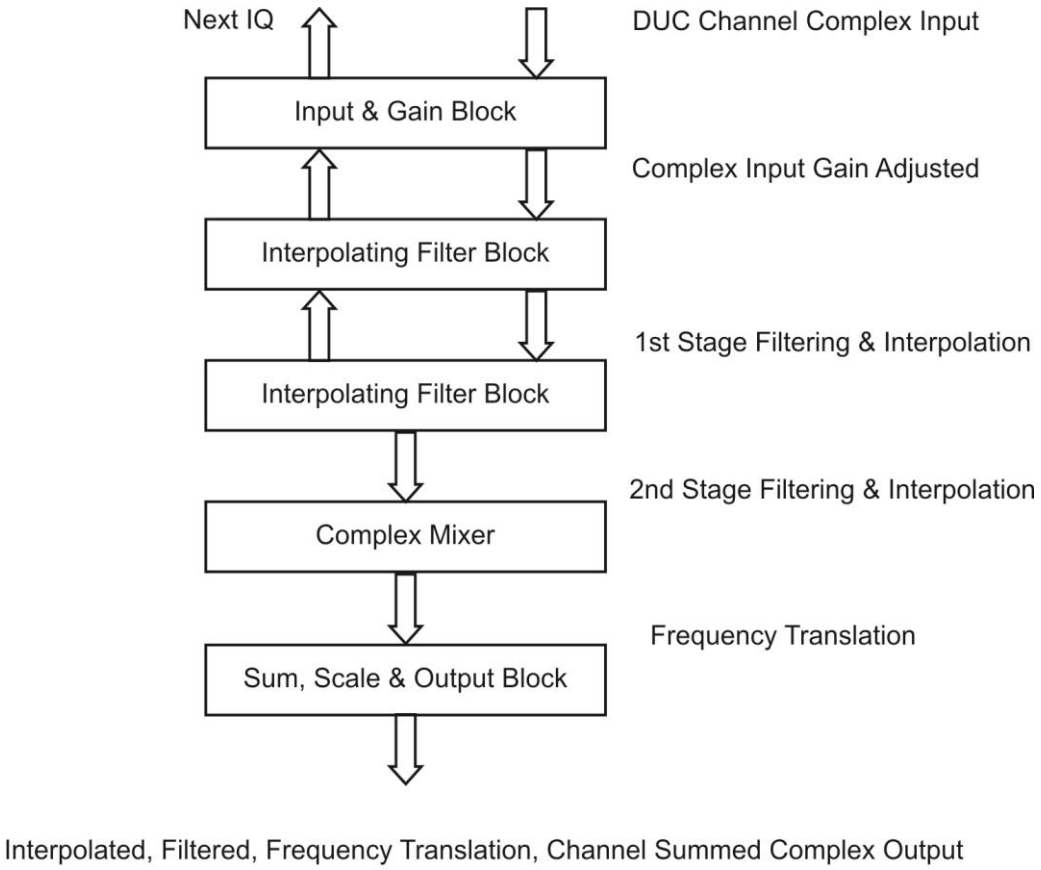

Figure 4.8: Two-stage DUC

4.6 IMPLEMENTATION OF QAM

Quadrature amplitude modulation (QAM) is equally analog as well as digital modulation method. It conveys two analog message signals or two digital bit streams by modulating the amplitudes of two carrier waves, using the amplitude shift keying (ASK) digital modulation scheme or amplitude modulation (AM) analog

modulation scheme. The two carrier waves, usually sinusoids are out of phase with each other by 90° and are thus called quadrature components. The modulated waves are merged and the consequential waveform is a blend of both phase shift keying (PSK) and amplitude shift keying (ASK). In the digital QAM case, at least two phases and at least two amplitudes are used. The PSK modulators are frequently designed using the QAM standard, but are not considered as QAM, as the amplitude of the modulated carrier signal is untiring. Quadrature phase shift keying (QPSK) waves have constant peaked sinusoidal wave, but the phase angle is different for the four various combinations of two bits. In 256 QAM, both amplitude and phase of the wave vary according to the 256 different combinations of eight bits.

The QAM is used widely as the modulation scheme for digital telecommunication systems. The QAM modulation is considered as an ideal solution to the optical fiber telecommunication bit rate amplification. The 16 QAM and 64 QAM can be optically emulated with a 3-path interferometer. A stable fine phase tuning system is required to enable a perfect phase control of the constellation. Both QAM and QPSK are modulation techniques used in IEEE 802.11 [wireless fidelity (Wi-Fi)], IEEE 802.16 (WiMax) and 3G [WCDMA/high speed downlink packet access (HSDPA)] wireless technologies. The modulated signals are then demodulated at the receiver, where the original digital message can be recovered. The use of adaptive modulation allows the wireless technologies to optimize throughput, yielding higher throughputs while covering the long distances.

Like all modulation schemes, the QAM holds data by changing some features of the carrier signal in response to the data signal. In the case of QAM, the amplitude of two signals, which is 90 degrees out-of-phase with respect to each other, is changed to modulate the data signal. The amplitude modulation of the two carriers in quadrature can

be evenly seen as both amplitude modulation and phase modulation of a single carrier. The phase modulation and phase-shift keying can be regarded as the special cases of QAM, where the magnitude of transformed signal becomes constant, with only the variable phase [102], [154]. The QAM is used in color television, Wi-Fi and WiMax, while QPSK finds its application in the code division multiple access (CDMA), cellular service, wireless local loop, and high speeds modems, etc.

4.6.1 QAM Transmitter

The figure 4.9 represents the model structure of a QAM transmitter, with a carrier frequency f_0 and the frequency response of the transmitter's filter H_t.

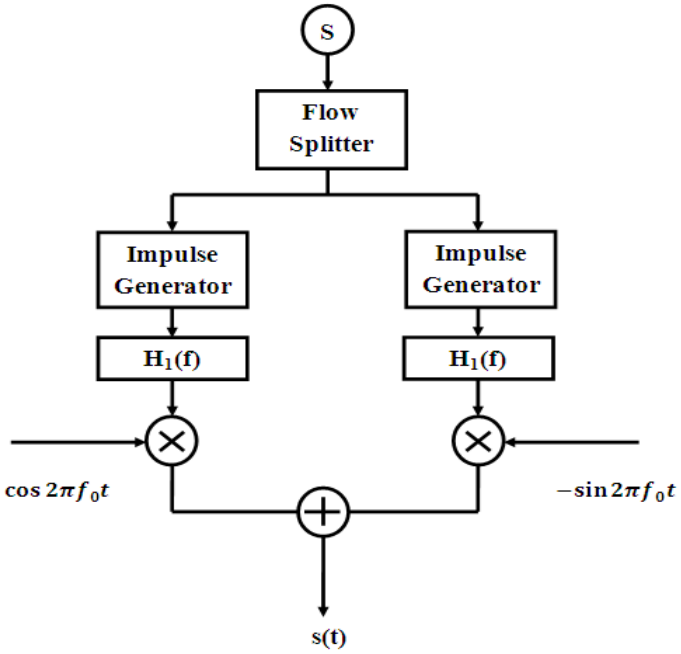

Figure 4.9: Model structure of QAM transmitter

First, the flow of bits to be transmitted is divided into two equal parts. This process generates two independent signals for the transmission. They are encoded separately, just like they were in an amplitude shift keying (ASK) modulator. Then one channel is multiplied by a cosine wave, while the other channel is multiplied by a sine wave. This way, there is a phase of 90 degrees between them. They are just added to the other and sent from end to end via the real channel [102].

4.6.2 QAM Receiver

The receiver merely performs the opposite operation of the transmitter. Its ideal structure is shown below in figure 4.10 with the receive filter's frequency response (H_r).

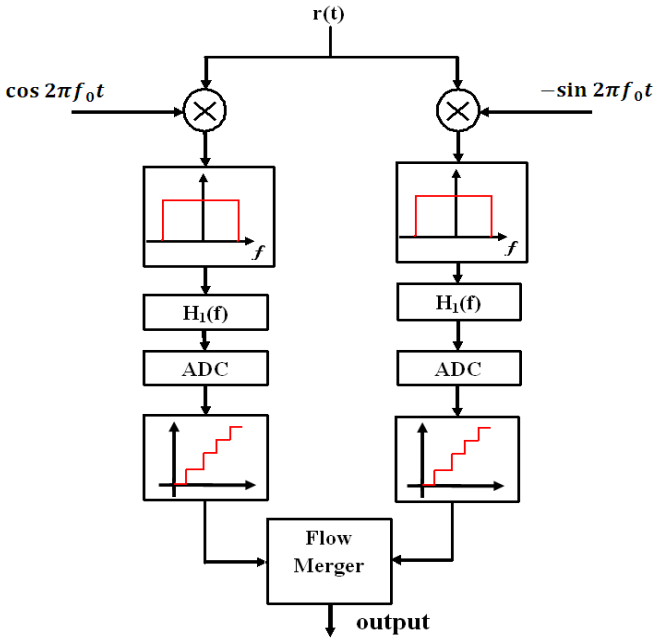

Figure 4.10: Model structure of QAM receiver

Multiplying by a cosine wave and by a low-pass filter, it is likely to extract the module in phase. Then, there is only an ASK demodulator, while the two flows of data are merged back. In practice, there is an unidentified phase delay among the transmitter and receiver that must be compensated by the synchronization of the receiver's local oscillator, i.e., the sine and cosine functions in the figure 4.10. In mobile applications, there will be an offset occurrence repeatedly in the relative frequency as well, due to the possible presence of a Doppler shift proportional to the relative velocity of the transmitter and receiver. Both the phase and frequency variations introduced by the channel must be compensated by appropriate fine-tuning of the sine and cosine components, which requires a phase reference, and it is typically accomplished using a phase locked loop (PLL) [102].

4.7 IMPLEMENTATION OF QPSK

The multi-stage filters have also been implemented for the QPSK based single-stage and double-stage interpolators and the results have been compared. Sometimes, this is known as quaternary PSK, quadric-phase PSK, 4 PSK or 4 QAM. Although the root concepts of QPSK and 4 QAM are different, the resulting modulated radio waves are precisely the same. In constellation diagram, the QPSK uses four points equally spaced around a circle, which employs four phases. And the QPSK can encode two bits per symbol, with coding in terms of the gray code to reduce the bit error rate (BER), sometimes misapprehended as twice the BER of binary phase shift keying (BPSK). The QPSK can be used either to double the data rate compared with a BPSK system, while maintaining the same bandwidth of the signal, or to maintain the data-rate of BPSK with reduced bandwidth requirement. In the latter case, the bit error rate (BER) of QPSK is precisely the same as the BER of BPSK.

Given that the radio communication channels are allocated by international agencies allowing a prescribed (maximum) bandwidth, the advantage of QPSK over BPSK becomes apparent. The QPSK sends twice the data rate in a given bandwidth compared to BPSK, at the same BER. In practice, the QPSK transmitters and receivers are more complicated than the BPSK transmitters and receivers. However with recent electronics technology, the cost penalty is temperate. As with QPSK, there are phase ambiguity problems at the receiving end, and the differentially encoded QPSK is often used in practice.

4.8 METHODOLOGY BASED ON SOFTWARE TOOLS

In this research work, the above discussed filters have been simulated in the Agilent's advanced design system (ADS). The ADS is the principal electronic design automation software for the radio frequency (RF), microwave and high speed digital applications. It

is being used by the foremost companies in wireless communication, networking, aerospace and defense industries. For microwave links, radar and satellite applications, ADS provides standard-based design and verification with wireless libraries on an integrated platform. The multi-stage filters have been implemented initially for the QAM and then for the QPSK based single-stage and double-stage interpolators; and the outputs have been compared. The suggested technique is not only applicable to interpolators, but also to the decimators as well.

4.9 RESULTS AND DISCUSSION

4.9.1 Simulation Results of 256 QAM Based Single-stage Interpolator

The block diagram of the 256 QAM based single-stage interpolator is shown in figure 4.11.

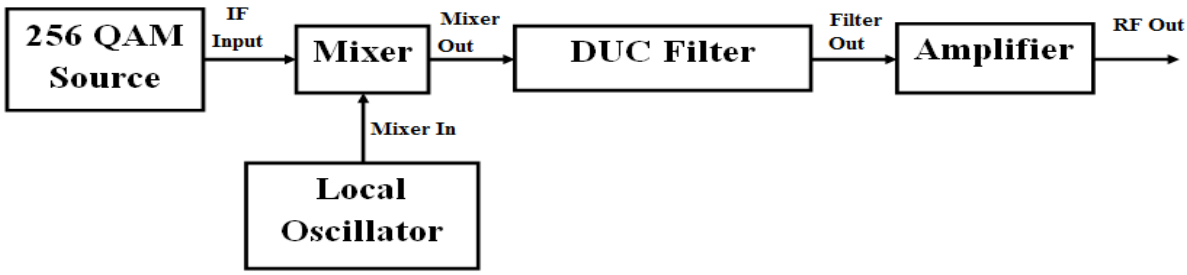

Figure 4.11: Block diagram of 256 QAM single-stage interpolator

The IF input is set at 0.988 GHz, and then the local oscillator frequency is mixed with IF frequency using the mixer in DUC block to get the RF output at 1.988 GHz. The spectrum of the generated signal for 256 QAM based single-stage interpolator is shown in figure 4.12 and the envelope power for 256 QAM based single-stage interpolator is presented in figure 4.13.

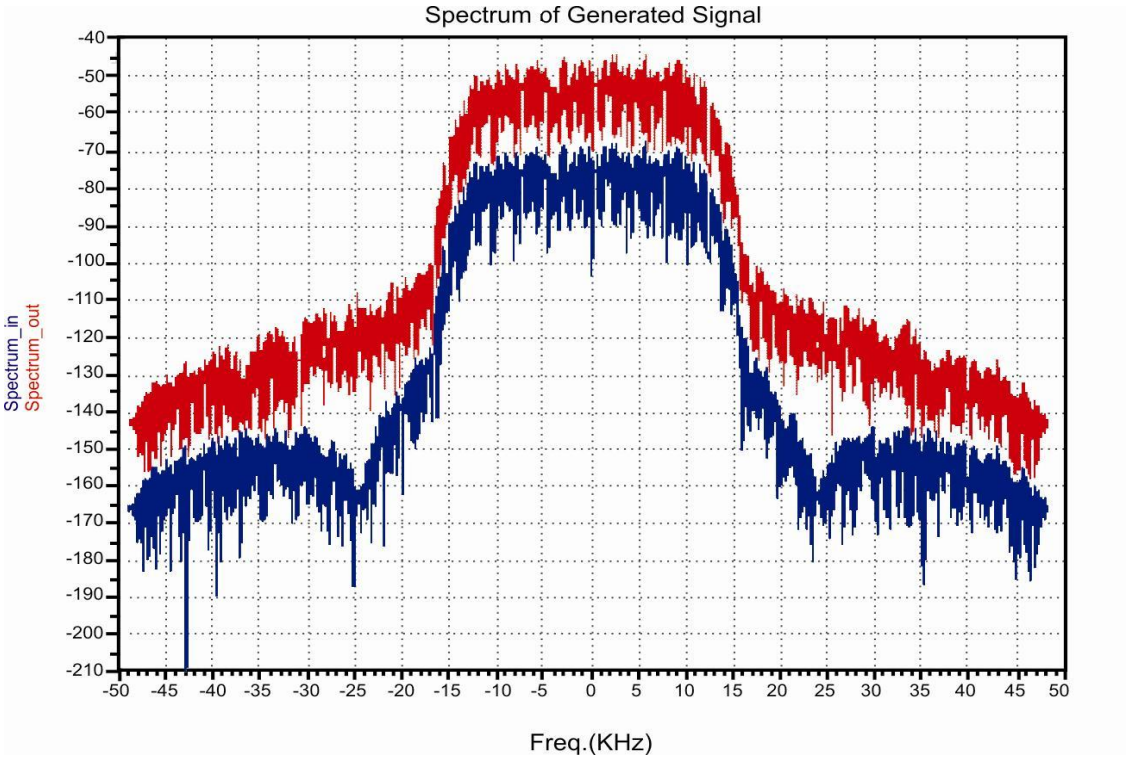

Figure 4.12: Spectrum of generated signal for 256 QAM single-stage interpolator

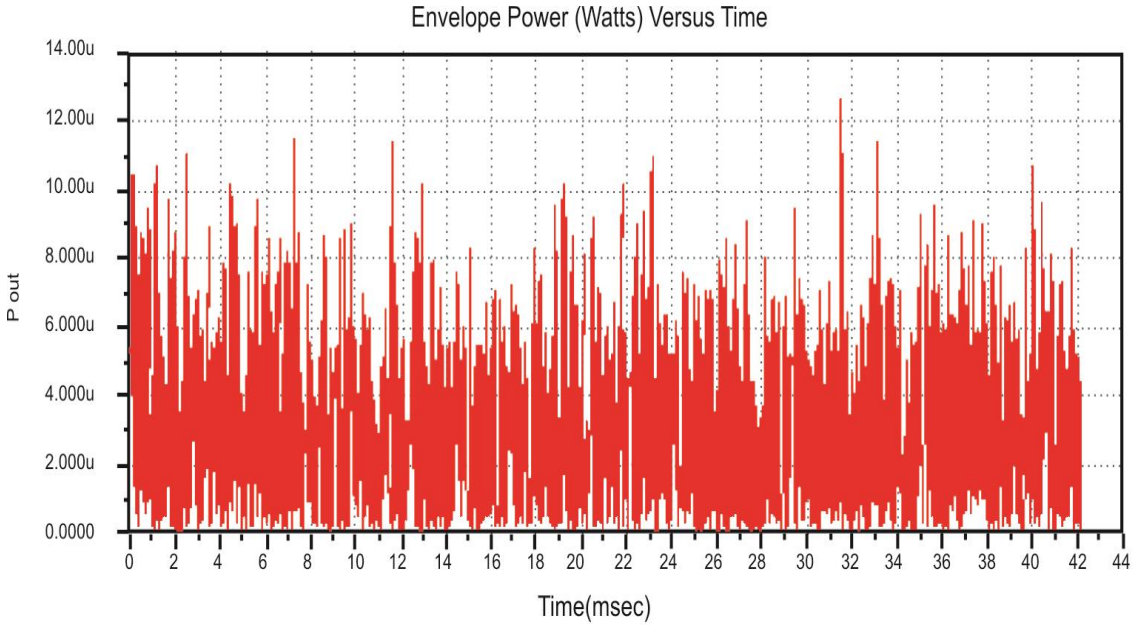

Figure 4.13: Envelope power for 256 QAM single-stage interpolator

The total output power comes out to be -25.19 dBm and peak to average power is 6.166 dB. The modulation signal trajectory and constellation for 256 QAM based single-stage interpolator is projected in figure 4.14.

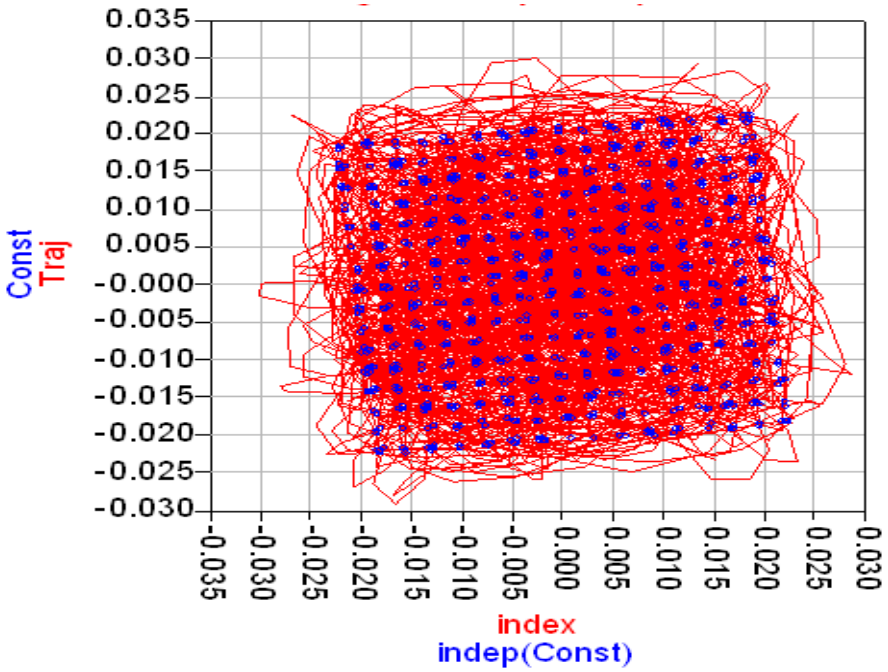

Figure 4.14: Modulation signal trajectory and constellation diagram for 256 QAM single-stage interpolator

The constellation diagram shows that, closer the points are to the origin, the less power it takes to send and accelerate the data rate (bit rate) at a fixed symbol rate. The percentage error vector magnitude comes out to be 0.240. The noise statistics calculated for the amplifier-mixer USB-filter amplifier configuration is compiled in the table 4.1.

TABLE 4.1

NOISE STATISTICS CALCULATED FOR 256 QAM AMPLIFIER-MIXER USB-FILTER AMPLIFIER CONFIGURATION

Paramenter	**Mix_In**	**Mix_Out**	**Filt_Out**	**RF_Out**
Power gain (dBm)	-32.137	-41.453	-39.784	-31.625
Cascade power gain (dB)	18.639	16.372	14.623	19.134
Noise figure (dB)	3.993	8.186	8.224	8.352

4.9.2 Simulation Results of 256 QAM based Double-stage Interpolator

The block diagram of the 256 QAM double-stage interpolator is plotted in figure 4.15.

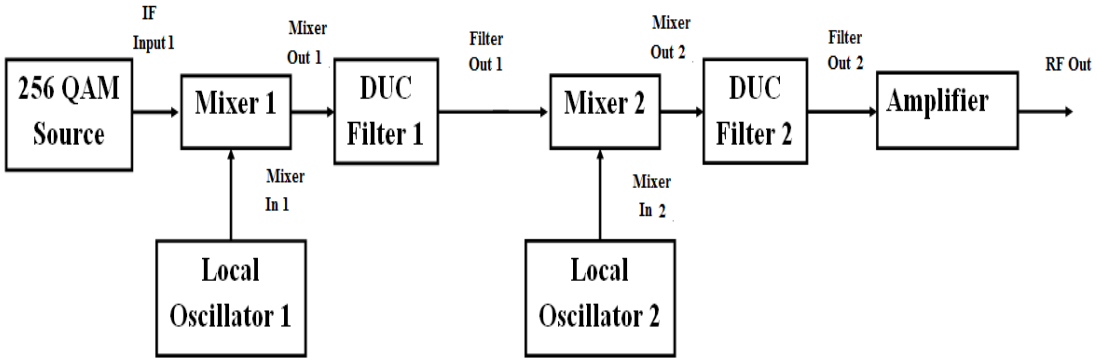

Figure 4.15: Block diagram of the 256 QAM based double-stage interpolator

The IF input is again set at 0.988 GHz, and then the 256 QAM based double-stage interpolator block is used to get the RF output at 1.988 GHz, which consists of two local oscillators and two mixers with two DUC filter blocks. The spectrum of generated signal for 256 QAM based double-stage interpolator is shown in figure 4.16. The envelope power for the 256 QAM double-stage interpolator is indicated in figure 4.17.

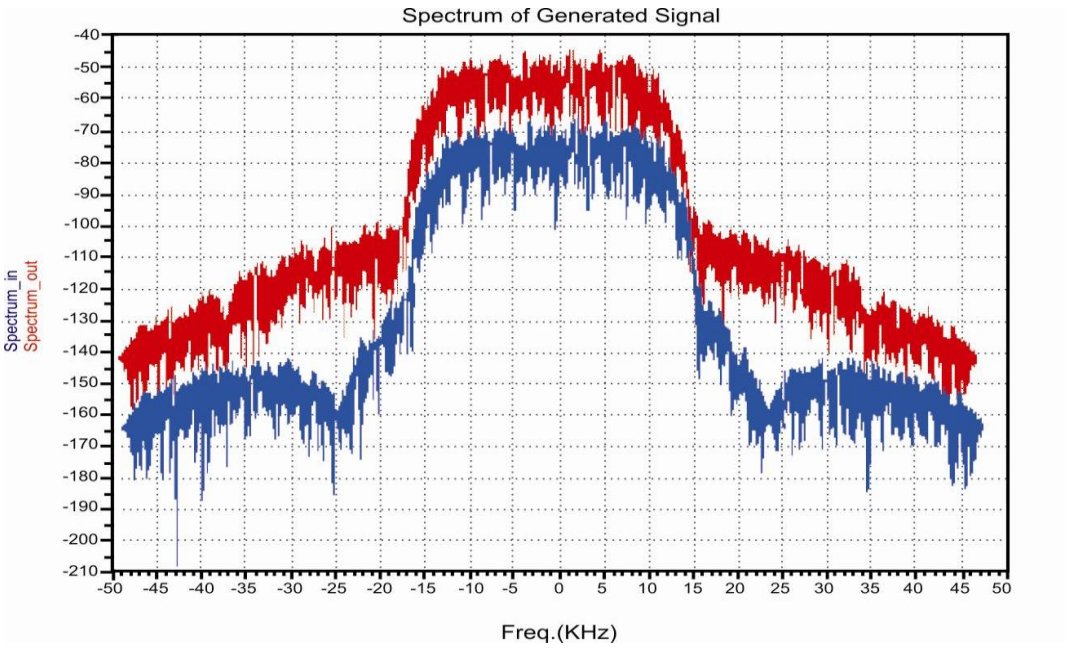

Figure 4.16: The spectrum of generated signal for 256 QAM double-stage interpolator

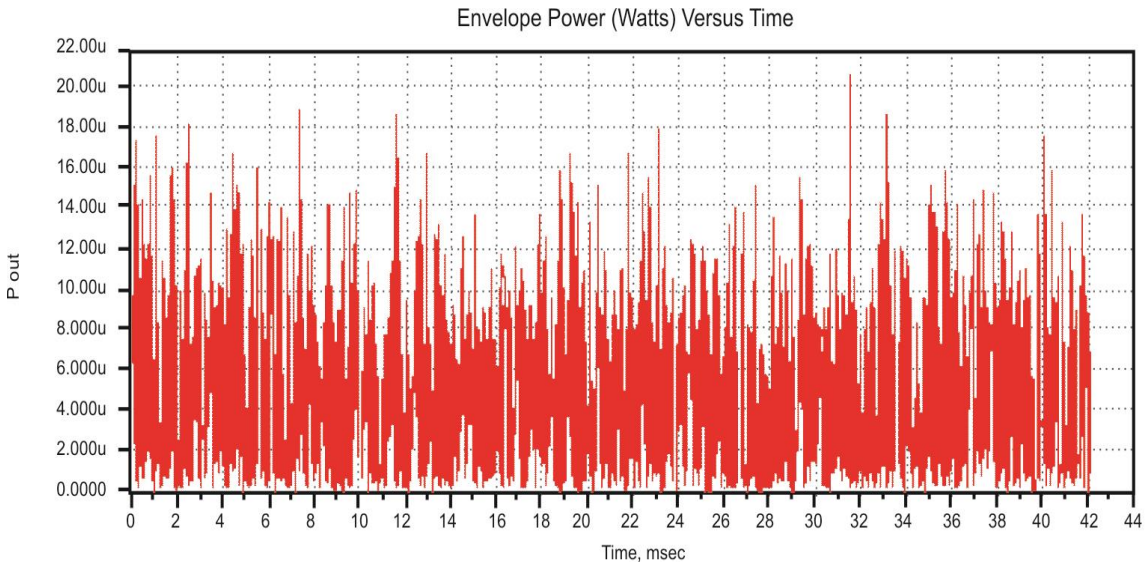

Figure 4.17: Envelope power for 256 QAM double-stage interpolator

The total output power comes out to be -22.96 dBm and peak to average power is 6.106 dB. The modulation signal trajectory and constellation for the 256 QAM double-stage interpolator is shown in figure 4.18.

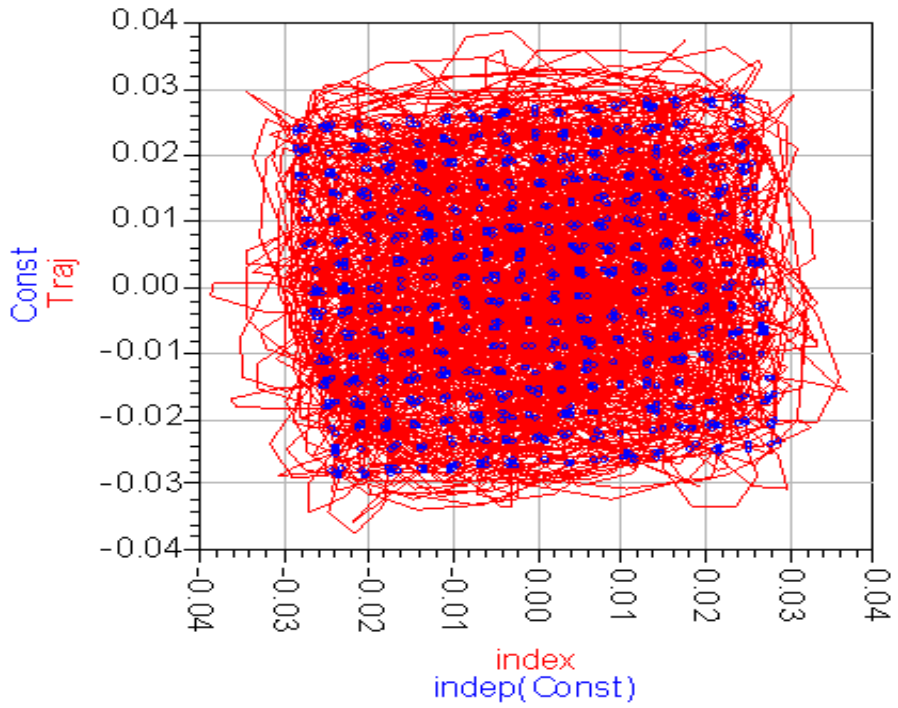

Figure 4.18: Modulation signal trajectory and constellation diagram for 256 QAM double-stage interpolator

From the constellation diagrams of 256 QAM based single-stage and double-stage interpolation filters in figure 4.14 and figure 4.18 respectively, it is evident that the double-stage interpolation filters take less power while the symbol synchronization improves, and the inter-symbol interference effects reduce drastically. The percentage error vector magnitude comes out to be 0.548. The noise statistics calculated for the 256 QAM double-stage amplifier-mixer USB-filter-amplifier-mixer USB-filter-amplifier configuration is compiled in table 4.2. From table 4.2, it is apparent that the double-stage QAM based interpolator is much better than the single-stage QAM based interpolator.

TABLE 4.2

NOISE STATISTICS CALCULATED FOR 256 QAM BASED AMPLIFIER-MIXER USB-FILTER-AMPLIFIER-MIXER USB-FILTER-AMPLIFIER

Parameter	Mix1_In	Mix1_Out	Filt1_Out	Mix1_In	Mix1_Out	Filt1_Out	RF_Out
Power gain (dBm)	-29.244	-35.765	-36.537	-26.648	-33.253	-34.473	-24.522
Cascade power Gain (dB)	20.856	12.342	13.183	23.241	16.416	13.229	23.614
Noise figure (dB)	3.874	6.986	6.991	6.996	7.012	7.115	7.152

4.9.3 Simulation Results of QPSK Single-stage Interpolator

The simulation diagram of the $\pi/4$ differential quadrature phase shift keying (DQPSK) single-stage interpolator is furnished in figure 4.19.

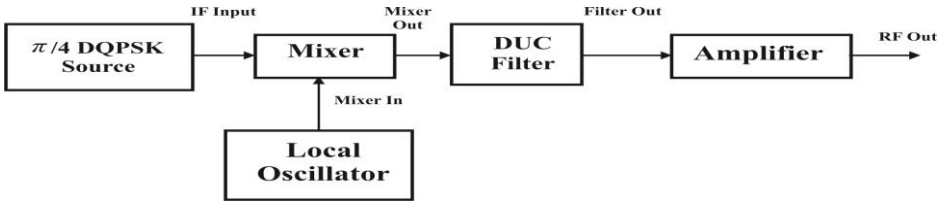

Figure 4.19: Simulation diagram of the $\pi/4$ DQPSK single-stage interpolator

The IF input is set at 0.988 GHz, and then the local oscillator frequency is mixed with IF frequency using the mixer in DUC block to get the RF output at 1.988 GHz. The spectrum of generated signal for $\pi/4$ DQPSK single-stage interpolator is shown in figure 4.20, whereas the modulation signal trajectory and constellation are plotted in figure 4.21.

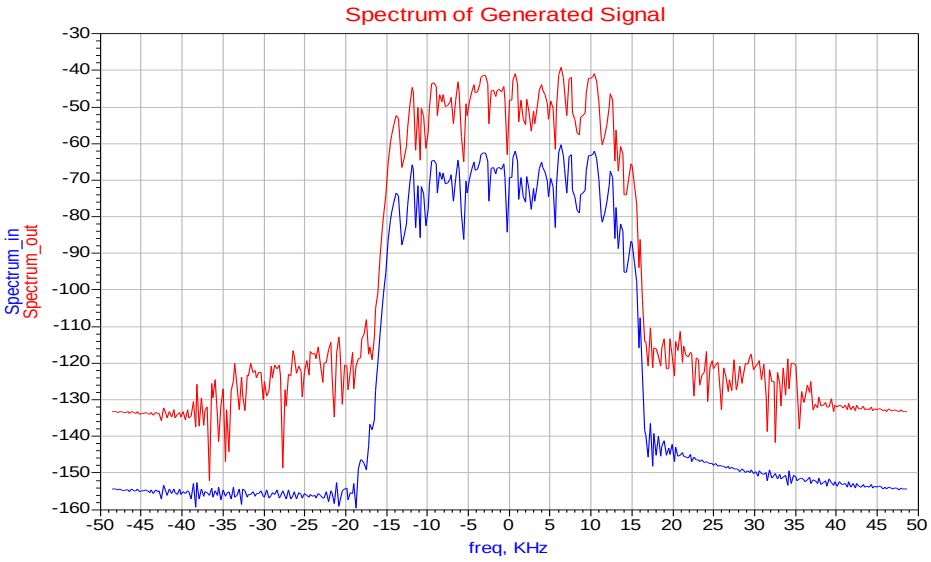

Figure 4.20: Spectrum of generated signal for $\pi/4$ DQPSK single-stage interpolator

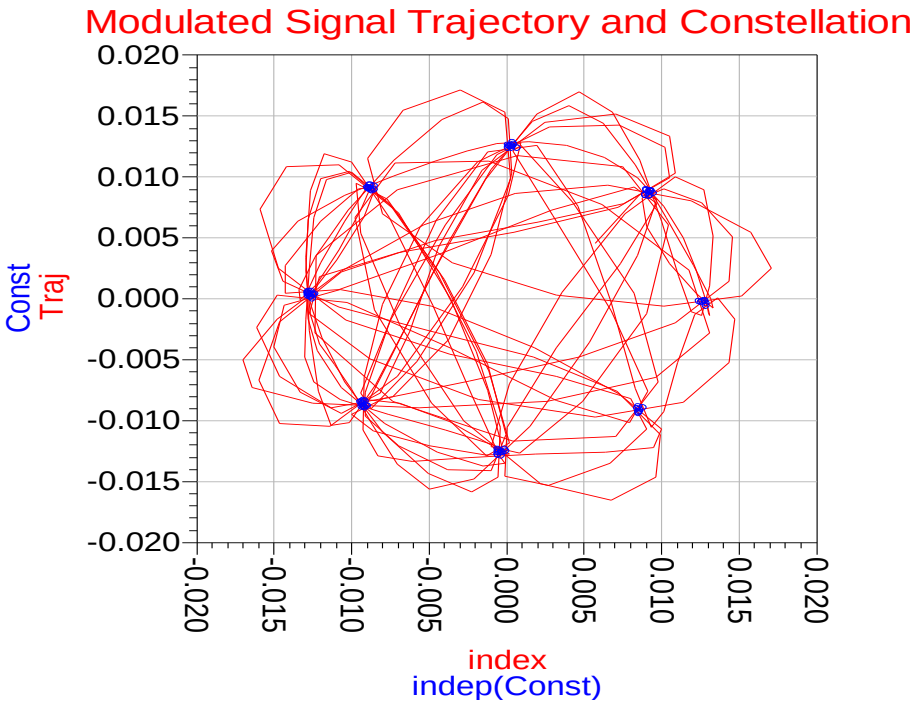

Figure 4.21: Modulation signal trajectory and constellation diagram for $\pi/4$ DQPSK based single-stage interpolator

The total output power comes out to be -28.09 dBm and peak to average power is 3.181dB in addition to the percentage error vector magnitude, which comes out to be 0.356. The noise statistics calculated for $\pi/4$ DQPSK amplifier-mixer USB-filter amplifier configuration is compiled in the table 4.3.

TABLE 4.3
NOISE STATISTICS CALCULATED FOR $\pi/4$ DQPSK AMPLIFIER-MIXER USB-FILTER AMPLIFIER CONFIGURATION

Paramenter	**Mix_In**	**Mix_Out**	**Filt_Out**	**RF_Out**
Power gain (dBm)	-30.133	-36.838	-37.854	-27.815
Cascade power gain (dB)	19.867	13.162	12.146	22.185
Noise figure (dB)	4.001	7.080	7.082	7.133

4.9.4 Simulation Results of QPSK Double-stage Interpolator

The simulation diagram of $\pi/4$ DQPSK double-stage interpolator is shown in figure 4.22.

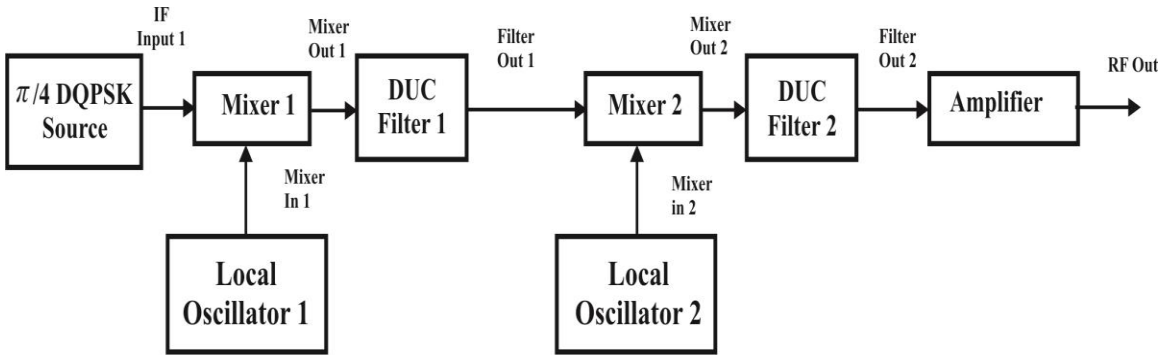

Figure 4.22: Simulation diagram of the $\pi/4$ DQPSK double-stage interpolator

The IF input is again set at 0.988 GHz and then the $\pi/4$ DQPSK based double-stage interpolator block is used to get the RF output at 1.988 GHz ,which consists of two local oscillators and two mixers with two DUC filter blocks. The spectrum of generated signal for $\pi/4$ DQPSK single-stage interpolator is demonstrated in figure 4.23 and the modulation signal trajectory and constellation are plotted in figure 4.24.

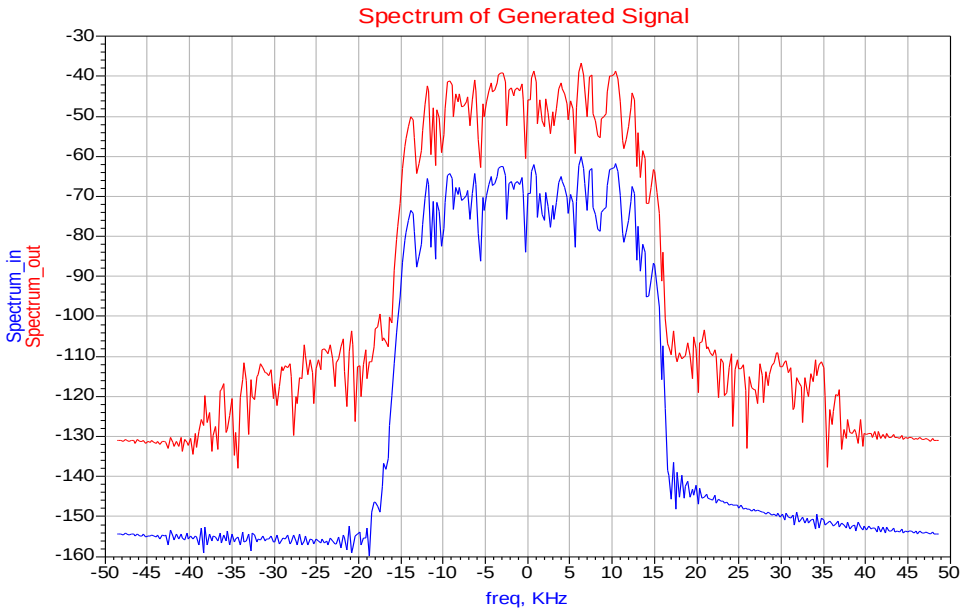

Figure 4.23: Spectrum of generated signal for π /4 DQPSK double-stage interpolator

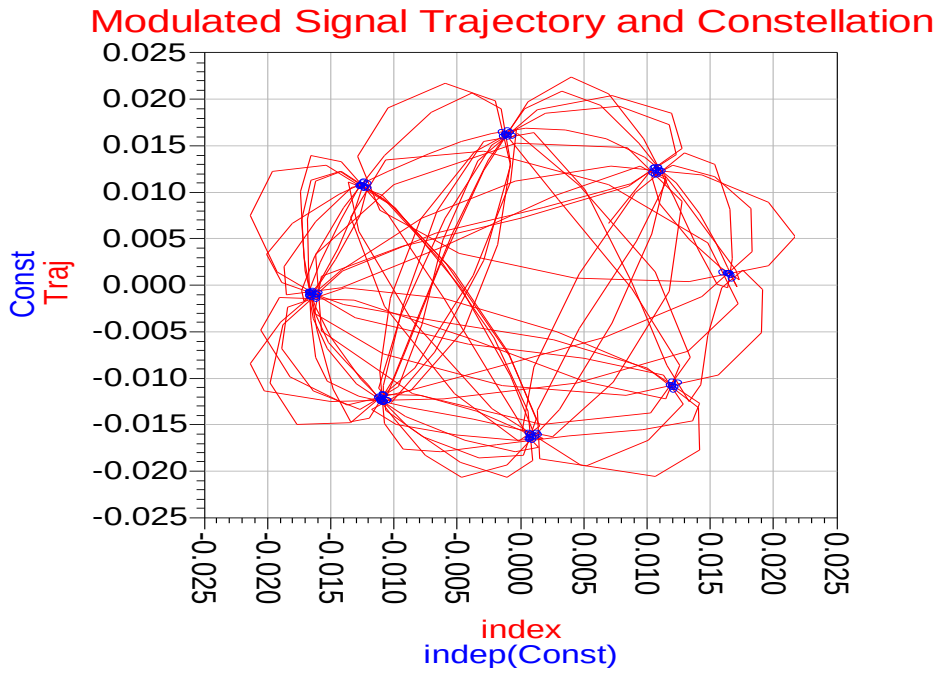

Figure 4.24: Modulation signal trajectory and constellation for π /4 DQPSK double-stage interpolator

The total output power comes out to be -25.84 dBm and peak to average power is 3.158 dB, while the percentage error vector magnitude comes out to be 0.817. The noise statistics calculated for $\pi/4$ DQPSK amplifier-mixer USB-filter-amplifier-mixer USB-filter-amplifier configuration is compiled in the table 4.4.

TABLE 4.4
NOISE STATISTICS CALCULATED FOR II/4 DQPSK AMPLIFIER-MIXER USB-FILTER-AMPLIFIER-MIXER USB-FILTER-AMPLIFIER CONFIGURATION

Parameter	Mix1_In	Mix1_Out	Filt1_Out	Mix1_In	Mix1_Out	Filt1_Out	RF_Out
Power gain (dBm)	-30.133	-36.856	-37.929	-27.869	-34.575	-35.591	-25.555
Cascade power Gain (dB)	19.867	13.144	12.071	22.131	15.425	14.409	24.445
Noise figure (dB)	4.001	7.079	7.082	7.126	7.216	7.217	7.241

From the above results, it is clear that the double-stage QPSK based interpolator is much better than single-stage QPSK based interpolator.

4.10 SUMMARY OF THE CHAPTER

The results in tables 4.1 and 4.2 demonstrate that the use of QAM based double-stage interpolators results in the increased power gain at the different points of interpolator, and the cascade power gain also escalates significantly. In addition, the total output power improves from -25.19 dBm to -22.96 dBm and the peak to average power improves 6.166 dB to 6.106 dB in the QAM based double-stage interpolation filter. Further, the noise figure is now much better than the QAM based single-stage interpolators.

Also, the results in tables 4.3 and 4.4 show that the use of QPSK based double-stage interpolators results in the increased power gain at the different points of interpolator, and the cascade power gain also increases substantially. Besides, the total output power improves from -28.09 dBm to -25.84 dBm and the peak to average power improves from 3.181 dB to 3.158 dB in the QPSK based double-stage interpolation filter. Moreover, the

noise figure is now much better than the QPSK based single-stage interpolators. Therefore, we can conclude that the double-stage interpolators are much more efficient and reliable than the single-stage interpolators. In this chapter, we have compiled the results only for 256 QAM and QPSK, but the proposed technique is also applicable to 16 QAM, 32 QAM, 64 QAM, 128 QAM. Hence, we can infer that by using the multi-stage interpolators, the performance of a system can be improved to a great extent. In the next chapter, the efficient implementation of multi-stage interpolator using the half-band filter has been proposed on FPGA platform.

CHAPTER 5

EFFICIENT IMPLEMENTATION OF MULTI-STAGE DIGITAL INTERPOLATORS USING HALF BAND FILTERS

5.1 INTRODUCTION

This chapter introduces an efficient implementation of the multi-stage configuration of digital interpolator using the half-band filters. The single-stage filters are efficient for the lower order interpolation factors, but for the high rate change (which is required in the modern digital communication systems like WCDMA, WiMax), the single-stage implementation does not give productive response. For these applications, a multi-stage implementation of the interpolator is desirable. However, the multi-stage implementation enhances the complexity of the overall system. In this chapter, it has been proposed that the complexity of multi-stage interpolators can be reduced considerably by using the half-band filters. All this has been done with the help of Agilent's advanced design software (ADS).

5.2 SINGLE-STAGE INTERPOLATION BY AN INTEGER FACTOR L

If the sampling rate is increased by an integer factor L, then the new sampling period T' is

$$\frac{T\prime}{T} = \frac{1}{L} \tag{5.1}$$

Here, the new sampling rate is $F' = LF$. This process of increasing the sampling rate of a signal *x(n)* by *L* implies that we must interpolate *L-1* new sample values between each pair of the sample values of *x(n)* [106].

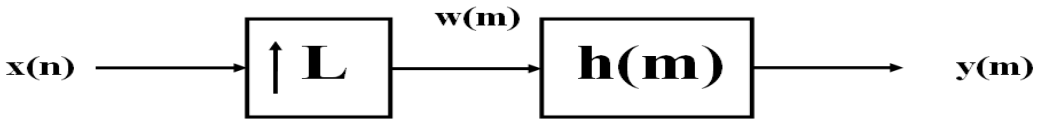

Figure 5.1: Block diagram of an Interpolator

The figure 5.1 represents the conceptual block diagram of an interpolator. For the interpolation by factor *L*, the *L-1* zero values have to be inserted between each pair of samples of *x(n)*, which results in the signal *w(m)*, such that

$$w(m) = \begin{cases} x\left(\frac{m}{L}\right), & m = 0, \pm L, \pm 2L, \dots \dots \\ 0, & otherwise \end{cases} \quad (5.2)$$

To recover the base-band signal of interest and to eliminate the unwanted higher frequency components, it is necessary to filter the signal *w(m)* with a digital low-pass filter, which approaches the ideal characteristic

$$\widetilde{H}\left(e^{j\omega\prime}\right) = \begin{cases} G, & |\omega'| \leq \frac{2\pi FT\prime}{2} = \pi/L \\ 0, & otherwise \end{cases} \quad (5.3)$$

If $H\left(e^{j\omega\prime}\right)$ denotes the frequency response of an actual filter that approximates the characteristic in equation (5.3), then

$$Y\left(e^{j\omega\prime}\right) = H\left(e^{j\omega\prime}\right)X\left(e^{j\omega\prime L}\right) \quad (5.4)$$

and, using the approximate value of $H\left(e^{j\omega\prime}\right)$ from equation (5.3), the equation (5.4) can be written as

$$Y\left(e^{j\omega\prime}\right) = \begin{cases} GX\left(e^{j\omega\prime}\right), & |\omega'| \leq \pi/L \\ 0, & otherwise \end{cases} \quad (5.5)$$

5.3 MULTI-STAGE IMPLEMENTATION OF AN INTERPOLATOR

The interpolator structures discussed above are the single-stage filters. These structures are efficient for the lower order interpolation factors. However for the higher order interpolation factors, the single-stage implementation can not be used. A multi-stage implementation [129], [141] of the interpolator is preferred in these applications. The interpolation factor *L* is factored into the product of integers i.e., $L=L_1L_2\dots\dots\dots L_K$ and applied as a cascade of *K* interpolators as shown in the figure 5.2. This structure is called the multi-stage interpolator.

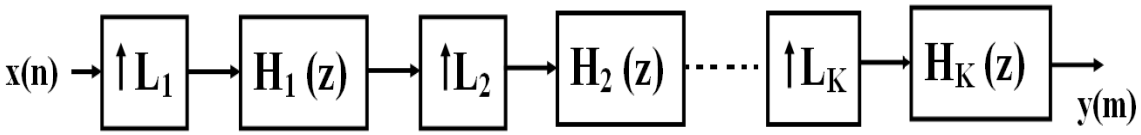

Figure 5.2: Multi-stage interpolator

The transfer function of the multi-stage interpolator is given as

$$H(z) = H_1(z)H_2(z^{L_1})H_3(z^{L_1L_2}) \dots \dots \dots H_K(z^{L_1L_2 \dots\dots L_{K-1}}) \quad (5.6)$$

The direct form as well as the poly-phase structures can be employed efficiently using the multi-stage implementation, which leads to less computational complexity in the implementation of an interpolator [150]. The multi-stage implementation of an interpolator results in the reduction of computational complexity, less storage requirement in the system, and the simplified filter design. Despite these advantages, the multi-stage implementation has following shortcomings

- The number of stages of the system can not be increased beyond a certain value because after that period, the complexity of the system starts increasing.
- Appropriate control structure is required to handle all the stages of the underlying system.

5.4 LTH BAND LINEAR PHASE FILTERS

The digital L^{th} band filters are a particular category of the digital filters, which are of special interest both in the single-rate and multi-rate digital signal processing. The common characteristic of the L^{th} band low-pass filters is that the cut-off angular frequency is located at π/L, and the transition-band is approximately symmetric around this frequency. In time-domain, the impulse response of an L^{th} band digital filter has zero valued samples at the multiples of L samples counted away from the central sample to the right and left directions. In reality, an L^{th} band filter has zero crossings at the regular distance of L samples, thus fulfilling the well known zero inter-symbol interference

property. Sometimes the L^{th} band filters are also called the Nyquist filters [150], [155]. The most significant structural improvement in the case of L=2 is the rudimentary benefit in applying L^{th} band filter, when every second coefficient in the transfer function is zero-valued. Due to the zero inter-symbol interference property, the L^{th} band filters are very important for the digital communication transmission/reception systems. These filters are also used as prototypes in constructing the critically sampled multi-channel filter banks. They are prevalent in the sampling rate alteration systems as well, where they are incorporated as decimation and interpolation filters in the single-stage and multi-stage systems. The filter transfer function $H(z)$ of such a filter can be expressed in the given form as

$$H(z) = \sum_{n=0}^{2K} h[n] z^{-n} \tag{5.7}$$

where, the filter length N is an odd number

$$N = 2K + 1 \tag{5.8}$$

As the filter exhibits linear phase characteristics, therefore the impulse response coefficients are symmetric

$$h[2K - n] = h[n] \qquad for\ n = 0,1,2, \dots\dots\dots\dots\dots 2K \tag{5.9}$$

The frequency response of a linear phase filter is expressible in the form of

$$H\left(e^{j\omega}\right) = e^{-jK\omega} H(\omega) \tag{5.10}$$

where $H(\omega)$ is the zero-phase frequency response, such that

$$H(\omega) = \frac{1}{L} + 2\sum_{n=1}^{K} h[K - n] \cos(\omega n) \tag{5.11}$$

The filter $H(z)$ is an L^{th} band filter, if the impulse response coefficients satisfy the following condition

$$h[K] = \frac{1}{L}, h[K \pm rL] = 0 \quad for\ r = 1,2, \dots\dots\dots\dots\dots K/L \tag{5.12}$$

the equation (5.12) defines the time-domain conditions for the L^{th} band filter. It was validated by Mintzer in [156] that the filters satisfying these time-domain conditions

defined by the equation (5.12), also satisfy the following condition in the frequency-domain

$$\sum_{r=0}^{L-1} H\left(\omega + \frac{2\pi r}{L}\right) = 1 \tag{5.13}$$

Apparently, this frequency-domain condition restricts the pass-band of low-pass filter to be smaller than π/L. The filter transition-band is symmetric around the angular frequency $\omega_c = \pi/L$, and the zero-phase frequency response has the value 0.5, which is evident in the figure 5.3. The pass-band and the stop-band edge frequencies ω_p and ω_s are symmetric around $\omega_c = \pi/L$, and are usually expressed in terms of the so-called roll-off factor ρ,

$$\omega_p = \frac{(1-\rho)\pi}{L}, and\ \ \omega_s = \frac{(1+\rho)\pi}{L} \tag{5.14}$$

where, ρ is a number in the range $0 < \rho < 1$. Obviously, the overall filter transition bandwidth, $\omega_s - \omega_p$ amounts to

$$\omega_s - \omega_p = \frac{2\pi\rho}{L} \tag{5.15}$$

It is also important to draw attention that the peak pass-band ripple δ_p is related to the peak stop-band ripple δ_s i.e.,

$$\delta_p = (L-1)\delta_s \tag{5.16}$$

Usually, the pass-band ripple δ_p is considerably smaller than this upper limiting value [156]-[157]. The L^{th} band filters exhibit a very attractive property, when these are used for interpolation. However, when the interpolation by L is performed with an L^{th} band filter, the original values of the input samples appear at the output without any distortion at the regular time intervals of L samples. It is well known that the interpolation by L consists of two operations: the up-sampling by L, and the low-pass filtering. Therefore, the samples $y[m]$ appearing at the interpolator's output are to be computed by convolving the up-sampled signal $\{x_u[m]\}$ with the low-pass filter impulse response $\{h[n]\}$, such that

$$y[m] = L\sum_{k}^{2K} h[k]x_u[m-k] \tag{5.17}$$

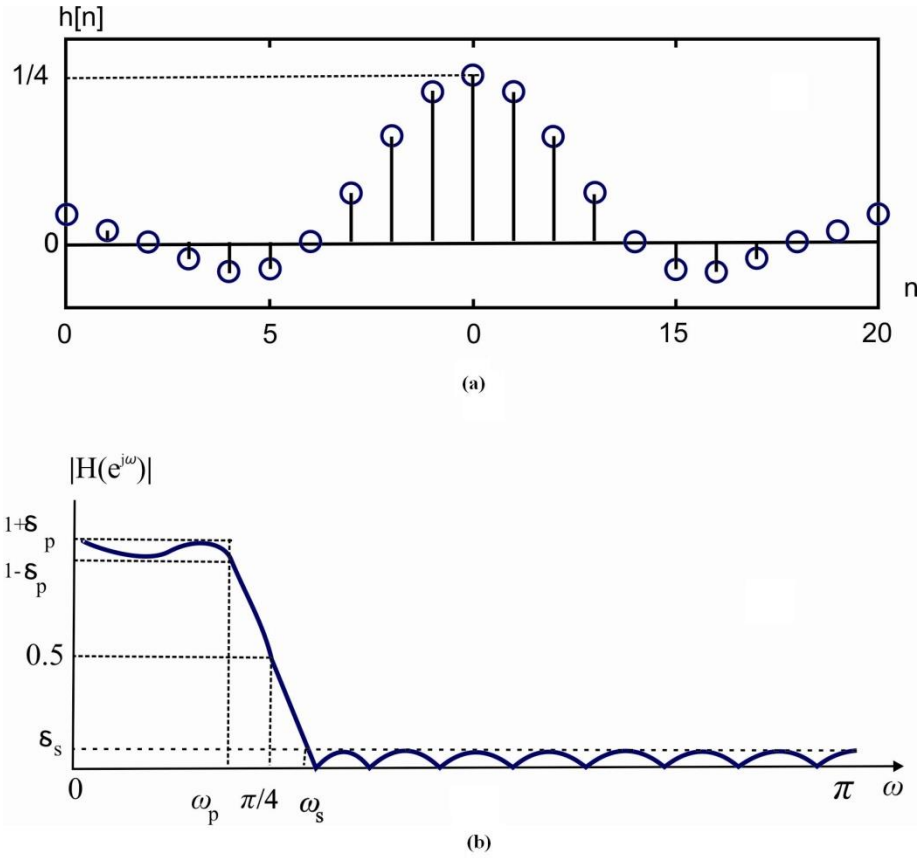

Figure 5.3: Illustration of the L^{th} band FIR filter properties for K = 10, L=4:

(a) Impulse response (b) Magnitude response

Thus, it is evident that the sequence $\{x_u[m]\}$ contains $L-1$ zero valued samples between each two non-zero samples, and the non-zero samples in $\{x_u[m]\}$ are those of the original signal $\{x[n]\}$. The ability of an L^{th} band filter to preserve the original sample values of $\{x[n]\}$ is due to the favorable coincidence that the zero crossings in the impulse response $\{h[n]\}$ and the non-zero sample values in $\{x_u[m]\}$ appear at the same regular time intervals. It can be demonstrated that what happens while computing the convolution [equation (5.17)] of the L^{th} band filter impulse response with up-sampling by factor L sequence. In figures 5.4(b) and 5.4(c), the sequence $\{x_u[m-k]\}$ has been shown for two distinct values of m (m_0 and m_1). The sequence in figure 5.4(b) illustrates the computation of the output sample, when the non-zero sample in $\{x_u[m-k]\}$ (marked with the arrow) coincides with the central sample of $\{h[k]\}$. In that case, the positions of remaining non-

zero samples of $\{x_u[m_0-k]\}$ coincide exactly with the zero crossings of the filter impulse response. Here, the convolution sum as shown in equation (5.17) has only single non-zero term, and therefore the sample value at the output is identical with that at the input. The figure 5.4(c) demonstrates the case when the position of the zero-valued sample of $\{x_u[m-k]\}$ coincides with the central sample of the impulse response. In that case, the positions of non-zero samples of $\{x_u[m_1-k]\}$ coincide with the non-zero samples of the filter impulse response. Consequently, the convolution sum calculated in equation (5.17) computes the output sample $y[n]$ as a non-zero interpolated value.

The figure 5.5 exhibits the overall results of the above interpolation. The sample values of the original signal are preserved as indicated using the thick lines. The in-between samples indicated using the thin lines are the interpolated values. Evidently, there is a delay of ten samples, as the filter length is kept $N = 21$.

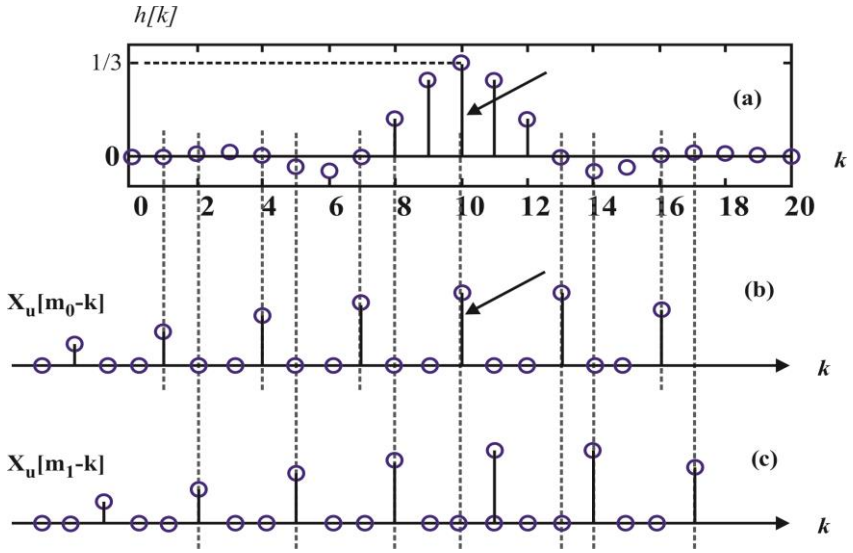

Figure 5.4: Interpolation with the L^{th} band filter with L=3

(a) Impulse response of the interpolation filter

(b) Sequence $\{x_u[m_0-k]\}$, $m_0 = 16$

(c) Sequence $\{x_u[m_1-k]\}$, $m_1 = 17$

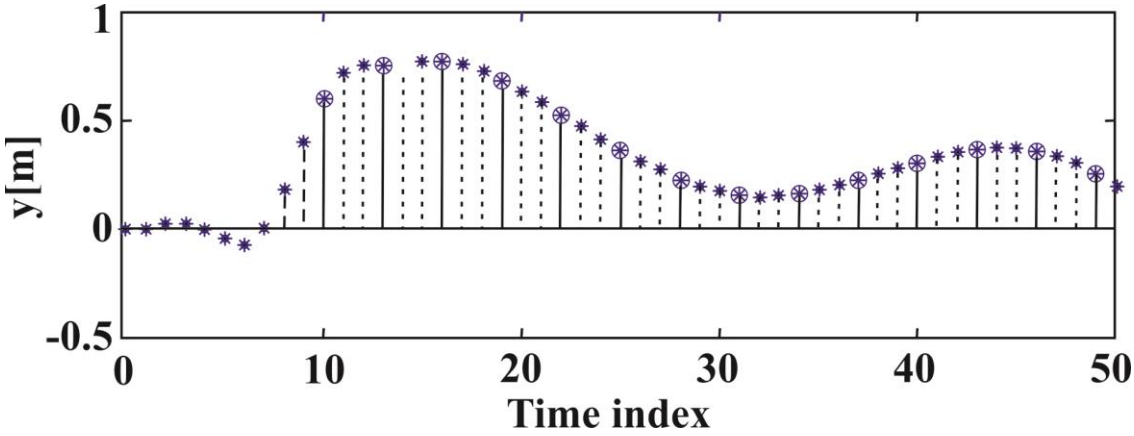

Figure 5.5: Interpolated signal: Thick lines with circles denote the samples that are identical with the input samples and thin lines with stars denote the interpolated sample values

5.5 HALF-BAND FILTERS

The half-band filter is an L^{th} band filter with L=2, and consequently the half-band filter divides the base-band signal into two equal sub-bands. In the linear phase half-band filter, half of the constants are zero-valued making the implementation very striking. The transfer function of a linear phase FIR half-band filter is presented as

$$H(z) = \sum_{n=0}^{2K} h[n]z^{-n} \qquad (5.18)$$

where K is odd, and the coefficients are symmetric with respect to the central coefficients $h[K]$, such that

$$h[2K-n] = h[n], \qquad for\ n = 0,1,2 \dots \dots \dots \dots 2K \qquad (5.19)$$

The filter length N is an odd number, which leads to

$$N = 2K+1, \qquad K = 1,3,5 \dots \dots \dots \dots \qquad (5.20)$$

In the case of a half-band filter, the time-domain condition defined by equation (5.12) for an L^{th} band filter becomes

$$h[K] = \frac{1}{2}, \quad h[K \pm 2r] = 0 \qquad for\ r = 1,2 \dots \dots \dots \dots K/2 \qquad (5.21)$$

The above expression shows that the odd-indexed coefficients in $\{h\ [n]\}$ are zero valued except for the central coefficient $h[K]$, which is equal to 1/2. As K is an odd number, the

impulse response begins and terminates with the non-zero samples, $h[0] \neq 0$, $h[2K] \neq 0$. The figure 5.6 illustrates the impulse response of a typical linear phase FIR half-band filter. The impulse response shown in figure 5.6 has the length of $N = 11$ samples, giving $K = (N-1)/2 = 5$. As the value of K should be odd, the filter length can be augmented by the multiple of four samples. Hence, we can choose $N = 7, 11, 15, 19 \ldots$and so on. In the frequency-domain, the zero-phase frequency response of a half-band filter showcases symmetry property

$$H(\omega) + H(\pi - \omega) = 1 \tag{5.22}$$

The above property follows from the general L^{th} band filter frequency-domain condition as given in equation (5.13), when reduced to $L = 2$. The frequency-domain condition given in equation (5.22) implies the symmetry of pass-band and stop-band characteristics with respect to the middle of base-band, $\omega_c = \pi/2$. The pass-band and stop-band edges are symmetric, and it may be shown as

$$\omega_s = \pi - \omega_p \tag{5.23}$$

Here, the peak pass-band and stop-band ripples are equal

$$\delta_p = \delta_s = \delta \tag{5.24}$$

In the pass-band, $H(\omega)$ approximates unity within the tolerances $[1-\delta, 1+\delta]$, and in the stop-band, $H(\omega)$ approximates zero within the tolerance $[-\delta, \delta]$, (as illustrated in figure 5.6). The value of $H(\omega)$ at $\omega_c = \pi/2$ is exactly 0.5, therefore

$$H\left(\frac{\pi}{2}\right) = 0.5 \tag{5.25}$$

An equi-ripple half-band filter can be designed by using the Parks-McClellan-Rabiner algorithm, but for a large N, this numerical procedure generates an error in the values of coefficients. A convenient method, called the half-band filter trick, is described by Vaidyanathan and Nguen in [110] and Saramaki in [153]. This method reduces the computational requirements by a factor of two. In the first step, a modified filter of about

half of the required length is designed using the Parks-McClellan-Rabiner program. In the second step, the desired half-band filter is created by simply inserting zeroes for the odd-index coefficients, scaling the even-index coefficients by 0.5, and by adding the central coefficient equal to 0.5. In this way, sufficient coefficient accuracy is achieved, and the design time is considerably alleviated.

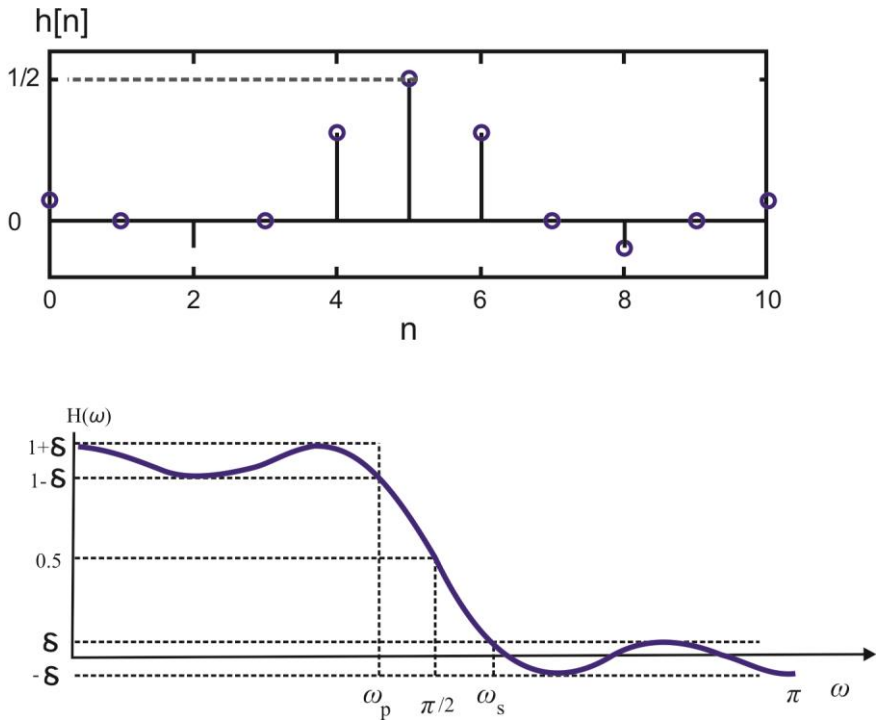

Figure 5.6: Linear phase FIR half-band filter: impulse response and zero-phase frequency response

5.6 EFFICIENT IMPLEMENTATION OF LINEAR PHASE HALF-BAND FILTERS

An important advantage of a linear phase half-band filter is the efficient implementation, which follows from the two favorable properties of the half-band filter's impulse response [86], [113] i.e.,

- The number of non-zero valued coefficients is nearly half of the filter length
- The non-zero coefficients exhibit symmetry property

For example, the efficiency of implementation can be verified by means of a linear phase

half-band filter of the length N = 11. Taking into account that every second coefficient is zero-valued except for the central coefficient, whose value is 0.5, the filter transfer function $H(z)$ can be written in the following form as

$$H(z) = h[0] + h[2]z^{-2} + h[4]z^{-4} + 0.5z^{-5} + h[6]z^{-6} + h[8]z^{-8} + h[10]z^{-10} \quad (5.26)$$

Introducing the coefficient symmetry property in the equation (5.26), the transfer function of the linear phase half-band filter becomes

$$H(z) = h[0] + h[2]z^{-2} + h[4]z^{-4} + 0.5z^{-5} + h[4]z^{-6} + h[2]z^{-8} + h[0]z^{-10} \quad (5.27)$$

It is convenient to represent $H(z)$ in terms of the two poly-phase components (sub-filters) $E_0(z)$ and $E_1(z)$, such that

$$H(z) = E_0(z) + z^{-1}E_1(z) \quad (5.28)$$

where, $E_0(z) = h[0] + h[2]z^{-1} + h[4]z^{-2} + h[4]z^{-3} + h[2]z^{-4} + h[0]z^{-5}$ (5.29)

and $E_1(z) = 0.5z^{-2}$ (5.30)

The coefficients in $E_0(z)$ are symmetric, and it may be given as

$$E_0(z) = h[0](1 + z^{-5}) + h[2](z^{-1} + z^{-4}) + h[4](z^{-2} + z^{-3}) \quad (5.31)$$

It can be seen that only three multiplication constants are needed to implement the linear-phase FIR half-band filter of the length N = 11. The value of central coefficient is 0.5 and therefore, this coefficient can be implemented with a binary shift. It is noteworthy fact that the sub-filter $E_0(z)$ preserves the coefficient symmetry of the original linear phase half-band filter transfer function. This property makes the linear phase FIR half-band filters highly efficient.

The linear phase FIR half-band filters when applied in factor-of-2 decimators and in factor-of-2 interpolators lead to very efficient realizations. The achievable efficiency of implementation is illustrated here by means of a factor-of-2 decimator and the linear phase half-band filter of N = 11. In figure 5.7 (a), the efficient configuration of a factor-of-2 decimator consisting of two poly-phase sub-filters $E_0(z)$ and $E_1(z)$ is showcased.

Since the linear phase half-band filter of $N = 11$ is to be considered, we implement the sub-filters $E_0(z)$ and $E_1(z)$ according to the equations (5.29), (5.30), and (5.31).

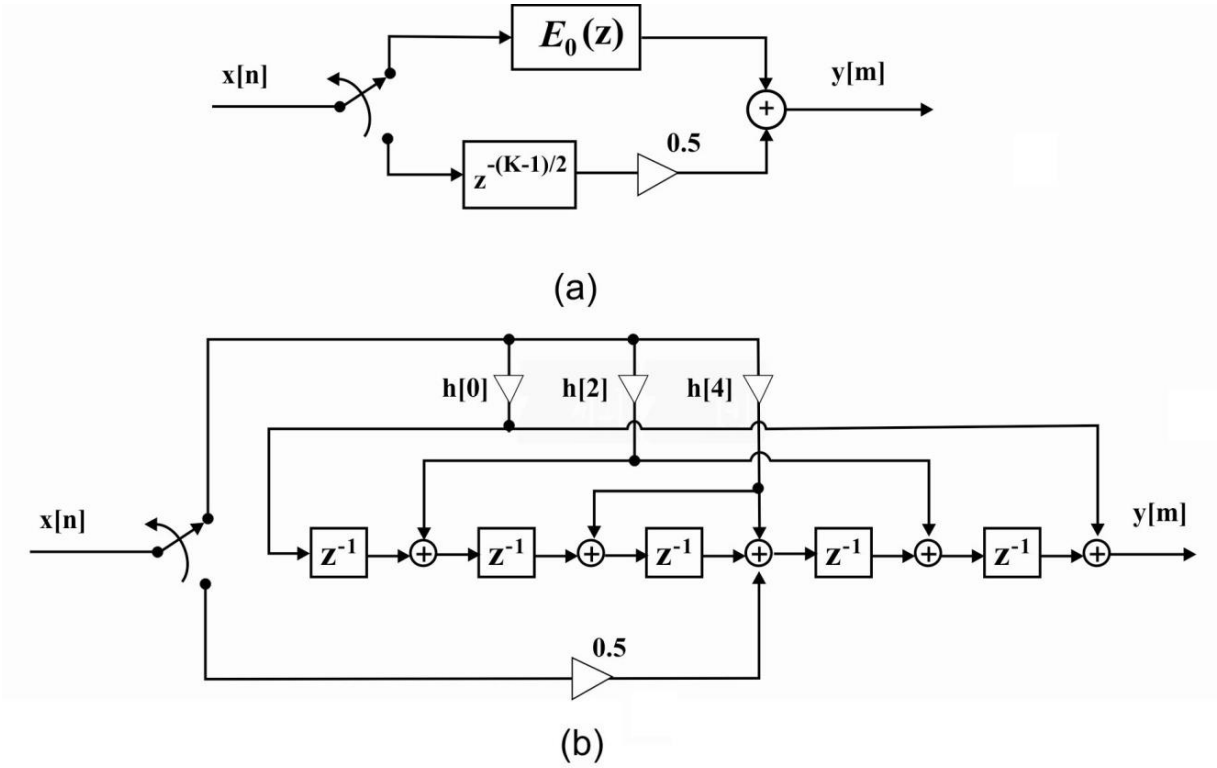

Figure 5.7: Efficient implementation of the linear phase FIR half-band filter (a) Poly-phase configuration. (b) Implementation scheme for $N = 11$

The favorable properties of $E_0(z)$ and $E_1(z)$ can be utilized to achieve very efficient implementation. The sub-filter $E_1(z)$ has only one term, as demonstrated in the equation (5.30). Here, $E_0(z)$ itself is of a linear phase with coefficient symmetry property, which is clearly evident in the equation (5.31). Thus, exploiting these favorable properties of $E_0(z)$ and $E_1(z)$, we arrive to the efficient implementation scheme for the overall decimator, which is illustrated in figure 5.7(b). The total number of multiplication constants is only three. Moreover, all the multiplications are to be evaluated at the rate of output signal. It is worth noticing that the delay chain in the figure 5.7(b) has only 5 elements. This is so because the memory-saving solution is used to reduce the total number of delays to $(N-1)/2$.

5.7 MINIMUM-PHASE AND MAXIMUM-PHASE HALF-BAND FILTERS

Assume that, in case of the L^{th} band filters, the half-band minimum-phase and the maximum-phase filters are $H_{\min}(z)$ and $H_{\max}(z)$ respectively, which are the spectral factors of the linear phase separable (factorizable) half-band filter $H(z)$, i.e.,

$$H(z) = H_{min}(z)H_{max}(z) \tag{5.32}$$

The zero-phase frequency response of $H(z)$ is non-negative, and the zeroes of $H(z)$ lying on the unit circle are the double-zeroes. The filter order should be even and twice of an odd number. The central coefficient of the impulse response has the value 0.5, and the remaining odd-indexed coefficients are zero-valued. Therefore, $H(z)$ is a half-band Nyquist filter with the non-negative zero-phase frequency response. Sometimes, this filter is called a valid half-band filter. An efficient design procedure exists for $H(z)$ based on the Remez exchange algorithm and on the above mentioned filter properties [153], [158]. Then the minimum-phase and maximum-phase filters $H_{\min}(z)$ and $H_{\max}(z)$ respectively share the transfer function zeroes of $H(z)$ in such a manner that $H_{\min}(z)$ selects zeroes inside the unit circle and one from each double-zero lying on the unit circle, whereas $H_{\max}(z)$ picks up zeroes outside the unit circle and one from each double-zero lying on the unit circle. The order of $H_{\min}(z)$ and $H_{\max}(z)$ is odd and half the order of $H(z)$. The impulse responses of $H_{\min}(z)$ and $H_{\max}(z)$ are reversed to each other. The figure 5.8 illustrates the impulse responses and the pole-zero plots of the minimum-phase as well as the maximum-phase filters. In the frequency-domain, the minimum-phase and maximum-phase half-band filters exhibit the power-symmetry properties. The term power-symmetry means that the squared magnitude response $|H(e^{j\omega})|^2$ have equivalent pass-band and stop-band ripples.

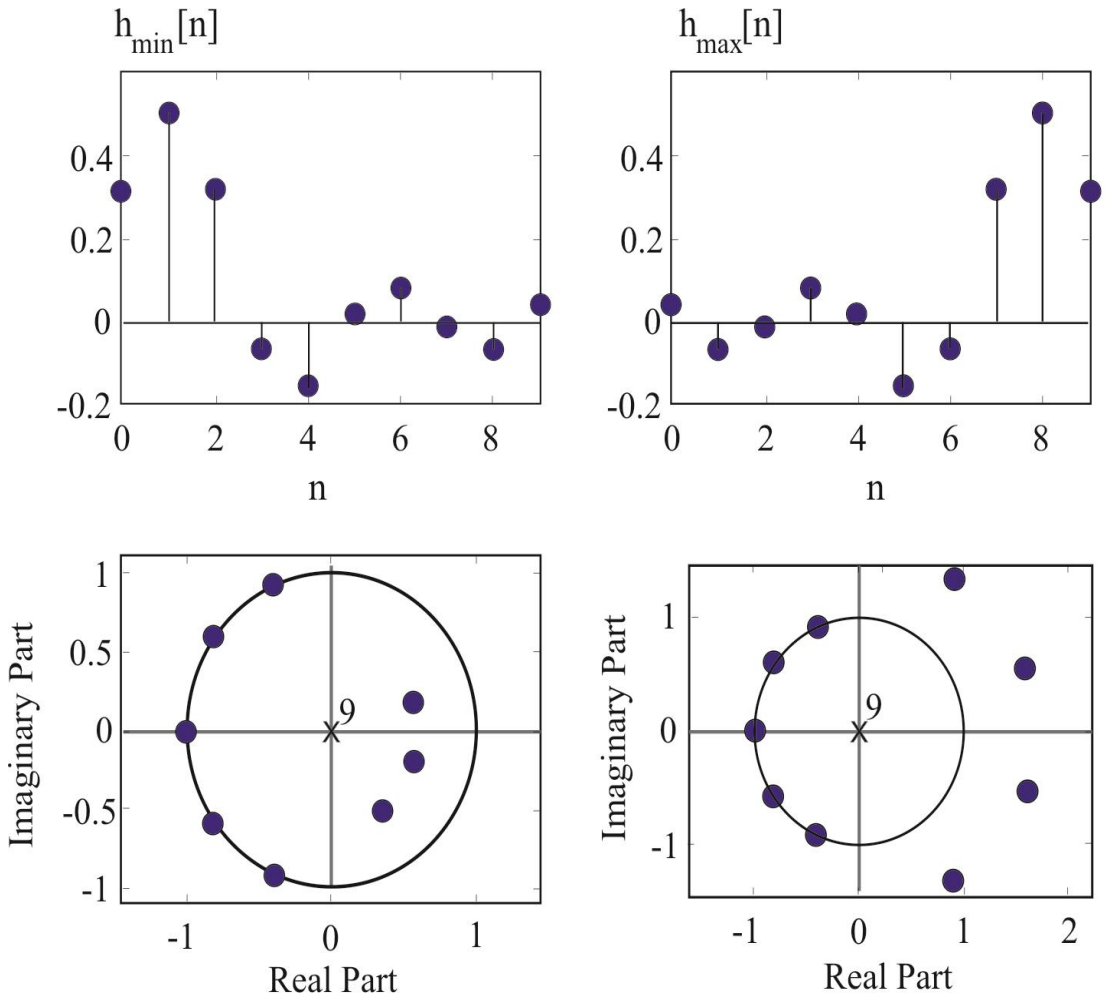

Figure 5.8: Impulse responses and pole-zero plots for minimum-phase and maximum-phase FIR half-band filters

The pass-band and the stop-band edge frequencies ω_p and ω_s are symmetric with respect to the central frequency $\omega_c = \pi/2$, and the frequency-symmetry property is expressed as

$$\omega_p + \omega_s = \pi \tag{5.33}$$

The magnitude response oscillates between $[1-\delta_p, 1]$ in the range $[0, \omega_p]$, and in between $[0, \delta_s]$ in the range $[\omega_s, \pi]$. The peak pass-band as well as stop-band ripples δ_p and δ_s are related as

$$\delta_p = 1 - \sqrt{1 - {\delta_S}^2} \tag{5.34}$$

thus, making certain that the pass-band and the stop-band ripples of $|H(e^{j\omega})|^2$ are equal.

When the minimal stop-band attenuation (a_S) and the peak pass-band ripples (a_p) are to be expressed in dB, their relation is determined as

$$a_p = 10log_{10}\left(1 + \frac{1}{10^{a_S/10}-1}\right) \quad (5.35)$$

The magnitude response at the middle of the base-band, at $\omega_c = \pi/2$, amounts to $1/\sqrt{2}$, i.e., the level of the magnitude response at this frequency is approximately 3 dB below the maximum. The magnitude response for both minimum-phase and maximum-phase filters is identical, and this unique characteristic is shown in figure 5.9. The minimum-phase and maximum-phase half-band filters are used in the multi-rate systems for constructing the filter banks with adequate reconstruction.

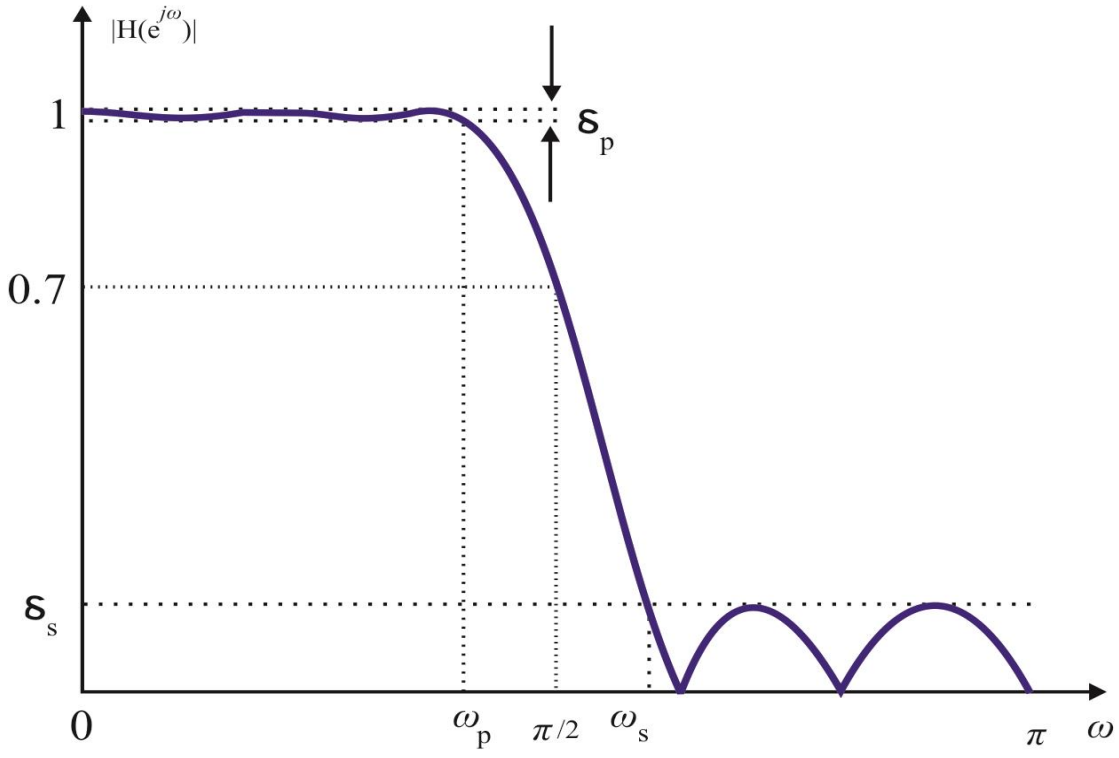

Figure 5.9: Typical magnitude characteristic of a half-band minimum-phase/maximum-phase FIR filter

5.8 RESULTS AND DISCUSSION

For implementing, the three-stage interpolator without using the half-band filter, we have connected three interpolators in the cascade which consists of two DUC filters (each filter interpolated by 2) followed by the pulse shaping circuit (interpolated by 2) with the following specifications

- **Sampling Frequency:** 46.08MHz
- **Bandwidth:** 5 MHz
- **Interpolation factor:** 8

The block diagram of the three-stage interpolator without half- band filter is shown in figure 5.10 and the output spectrum of three-stage interpolator without the half-band filter is illustrated in figure 5.11.

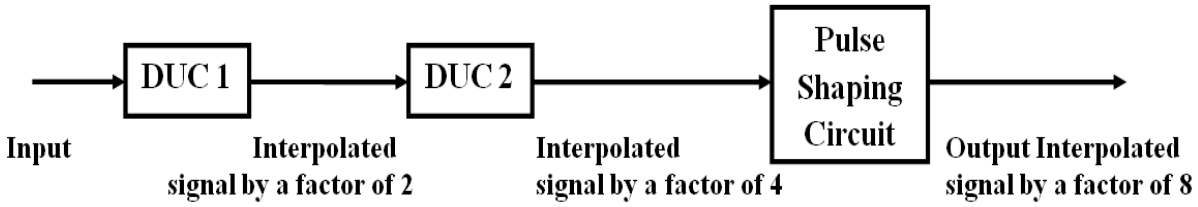

Figure 5.10: Three-stage interpolator without half-band filter

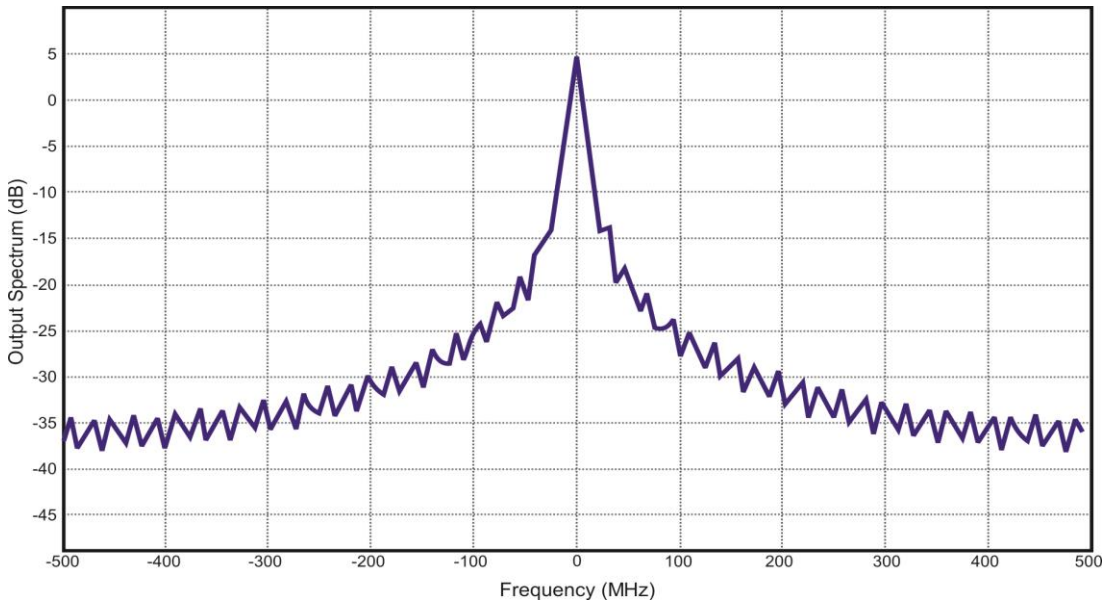

Figure 5.11: Output spectrum of three-stage interpolator without half-band filters

To implement, the three-stage interpolator using the half-band filter, we have connected three interpolators in cascade, which consists of the pulse shaping circuits (interpolated by 2) followed by the two half-band filters (each half band filter interpolated by 2) with the following specifications

- **Sampling Frequency** = 46.08MHz
- **Bandwidth** = 5 MHz
- **Interpolation factor** = 8

The block diagram of the three-stage interpolator with half-band filter is plotted in figure 5.12.

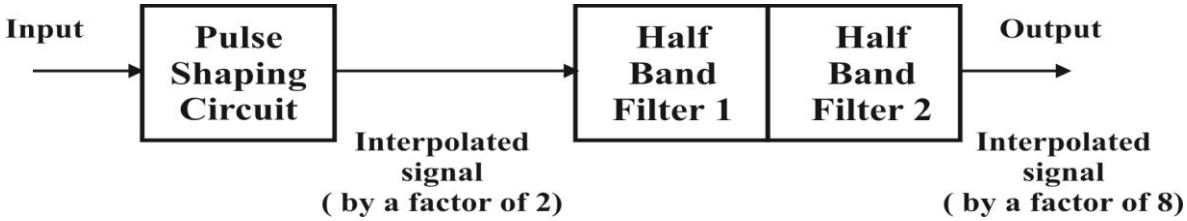

Figure 5.12: Three-stage interpolator using half-band filter

The output spectrum of three-stage interpolator with the half-band filter is shown in figure 5.13.

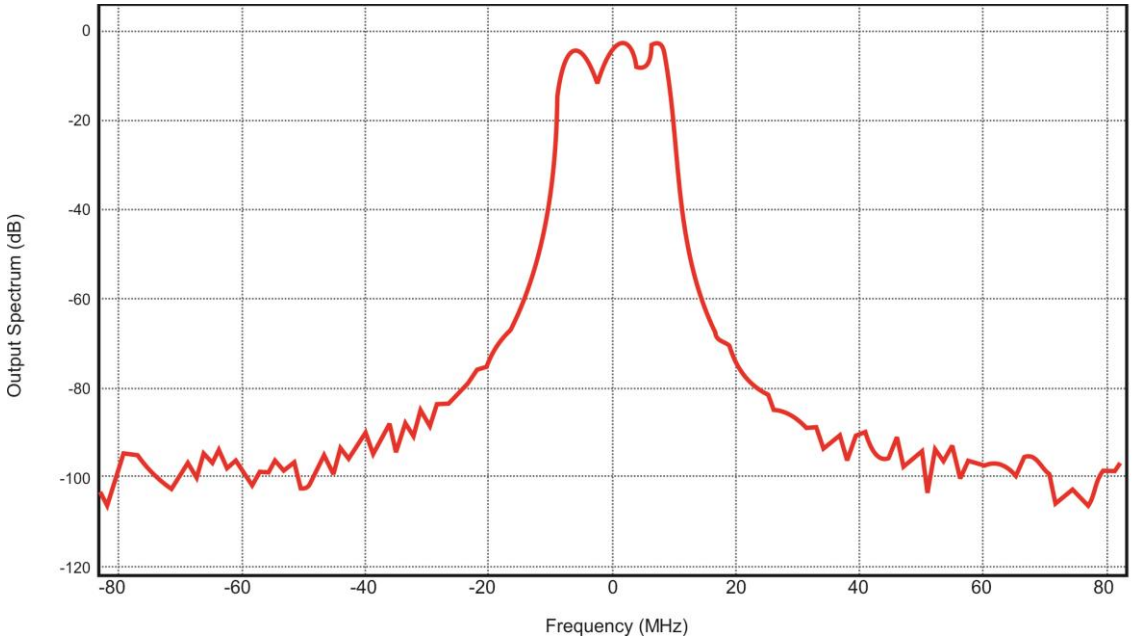

Figure 5.13: Output spectrum of three-stage interpolator with half-band filters

The simulation results manifest that the implementation of single-stage interpolator leads to the stop-band edge frequency of 14.6 MHz; but for the three-stage interpolator, this stop-band edge reduces to 6.7 MHz with sharper frequency response in the stop-band region. Therefore, the presented three-stage interpolation gives comparatively a sharper transition relative to the single-stage interpolation. Further, the stop-band edge frequency reduces to 3.7 MHz due to the use of half-band filters for the three-stage interpolators. Additionally, the reduction in the number of multipliers using the multi-stage interpolators has been shown in the table 5.1.

TABLE 5.1
NUMBER OF FILTER TAPS AND MULTIPLIERS REQUIRED FOR SINGLE-STAGE AND THREE-STAGE INTERPOLATOR USING HALF-BAND FILTERS

Bandwidth (MHz)	Single-stage interpolator (multipliers)	Three-stage interpolator (multipliers)		Comparison
		Without half-band filters	With half-band filters	
20	58	58	58	100%
15	112	87	76	68%
10	227	112	82	36%
7.5	456	135	96	21%
5	374	106	68	18%

The table 5.1 indicates the number of filter taps as well as the multipliers required and also the reduction in the number of multipliers for the three-stage interpolator using the half-band filter. This table also outlines that the number of multipliers essential for the three-stage interpolator using the half-band filters is less than that of the number of multipliers necessary for the single-stage interpolators and three-stage interpolator without using the half-band filters. For the channel bandwidths of 20 MHz, 15 MHz, 10 MHz, 7.5 MHz and 5 MHz, the three-stage filtering without the half-band filters can be realized using 100%, 78%, 49%, 30% and 28% of multipliers respectively, which are essential for the single-stage filtering. At the same time, the three-stages filtering with the

half-band filters can be realized using 100%, 68%, 36%, 21% and 18% of multipliers respectively, which are essential for single-stage filtering. In comparison to the single-stage filtering, the three-stage filtering (with and without half-band filters) with less number of multipliers does not result in the deterioration of power spectral density (PSD) at the edge of channel bandwidth. Thus, it is evident that the half-band filters are beneficial in designing the effective and efficient multi-stage digital interpolators from the implementation point of view.

5.9 SUMMARY OF THE CHAPTER

The primary requisite of digital communication system is the efficiency of the underlying system, which can be enriched to a large extent using the multi-stage interpolators, as it is apparent from the outputs. We can further improve the efficiency by cascading more and more interpolators, but it should be kept in mind that the system becomes more and more complex as the number of stages increases. This complexity can be reduced drastically by using the half-band filters. The results manifest that the number of multipliers essential for the three-stage interpolator using the half-band filters is less than that the number of multipliers necessary for the single-stage interpolators and three-stage interpolator without using the half-band filters. However, the overall results demonstrate that the efficacy of the proposed scheme over the conventional approaches.

CHAPTER 6

CONCLUDING REMARKS AND FUTURE SCOPE

6.1 CONCLUDING REMARKS

In the presented research work, the efficient structures of the digital interpolators have been proposed using the field programmable gate arrays (FPGA) technology. The different techniques have also been presented for improving the performance of digital interpolators at the different sampling rates.

First, the cascaded integrator comb (CIC) filters with compensation have been proposed on FPGA for the specifications of digital up-converter (DUC) filters for the different wireless standards. The results depict that the CIC filters reduce the implementation cost, as its implementation can be performed without the use of multipliers; which leads to relatively fast response as compared to the conventional approaches. But, the pass-band droop present in the CIC filters limit the scope of its practical applications. On the contrary, the compensation technique ensures the improved pass-band response of the CIC filter, as the normalized frequency reduces by approximately 74%, but at the cost of approximately 34% increased hardware requirements as well as computational complexity. However, the corresponding results indicate that the planned digital interpolator structures can be used very effectively and efficiently in the design and development of the very large scale integration as per the industrial standards for communication systems. The CIC filters are unsuitable for those practical applications, where the fractional delay is stringently required in the asynchronous systems. As an interesting remedy for this problem, the Farrow filter structures are broadly investigated in the literature and are extensively used in practice for attaining the fractional delay.

Next, the designing and investigation of interpolation filters using the Farrow filter structures have been carried out with the help of Lagrange interpolation technique, as it provides exact shaping to the interpolated signal as of the input signal. Subsequently, the Altera DSP builder advanced blockset has been used for designing the projected Farrow filter structures. It may be concluded from the presented results that even though the cubic Lagrange interpolation method provides somewhat less smooth interpolated signal as compared to the quadratic Lagrange interpolation method, but the spectrum of interpolated signal in former case has very less ripple contents when compared to the quadratic Lagrange interpolation spectrum. The ripple level reduces by approximately 58%, and the computational burden also decreases by approximately 42% in the case of multipliers and approximately 18% in case of additions for the cubic Lagrange interpolation approach. Hence, the Farrow filter structures for the cubic Lagrange interpolation are found to be the optimum solution for the designing of interpolation filters. In addition, the modified Farrow filter structure for the cubic Lagrange polynomial interpolation has also been proposed for the comparison between the conventional Farrow structures and the modified Farrow structures in case of the resource utilization. It can be inferred from the results/research outcomes that the number of look-up tables (LUTs) for the proposed design decreases by approximately 2.95%, the number of multipliers reduces by approximately 7.81% and the memory bits decrease by approximately 2.25%. However, the proposed structure manifests its superiority in terms of the spectral re-growth and the less area requirement when implemented on the FPGA platform.

Further, the simulation results have been compiled for the single-stage interpolation filters. Though the single-stage filters are efficient for the lower order interpolation factors, but for the high rate change, the single-stage design does not offer effective response. For such scenarios, the multi-stage realizations of the interpolators are

proposed. The efficient implementation configurations of digital interpolation systems for the up-sampling of QAM/QPSK based multi-stage digital interpolators have been investigated. The total output power improves by approximately 9% and the peak to average power improves by approximately 1% in the QAM based double-stage interpolation filter. The total output power improves by approximately 8% and the peak to average power improves by approximately 0.7% dB in the QPSK based double-stage interpolation filter. The presented technique is not only applicable to the 256 QAM and QPSK, but also appropriate for the 16 QAM, 32 QAM, 64 QAM, 128 QAM digital modulation schemes. Thus, it is apparent that by using the multi-stage interpolators, the performance of underlying digital communication system can be enhanced to a considerable degree. Unfortunately, the multi-stage implementation boosts the computational as well as implementation complexity of the overall communication system design. However, this complexity can be reduced significantly by using the half-band filters.

Therefore, the efficient designing of multi-stage digital interpolators using the half-band filters has also been proposed. It is evident from the simulation results that for the three-stage interpolator, this stop-band edge alleviates by approximately 54% with the sharper response in the stop-band region as compared to the single-stage interpolator. Consequently, the stop-band edge frequency reduces by approximately 74% because of the use of half-band filters for the three-stage interpolators. Therefore, the presented three-stage interpolation method exhibits a sharper transition.

For the channel bandwidths of 20 MHz, 15 MHz, 10 MHz, 7.5 MHz and 5 MHz, the three-stage filtering without the half-band filters can be realized using 100%, 78%, 49%, 30% and 28% of the multipliers respectively, which are essential for the single-stage filtering. However, the three-stages filtering with half-band filters can be realized using

100%, 68%, 36%, 21% and 18% of the multipliers respectively, which are essential for the single-stage filtering. In comparison to the single-stage filtering, the three-stage filtering (with and without half-band filters) with less number of multipliers does not result in the degradation of power spectral density at the edge of channel bandwidth. Hence, it can be concluded that the half-band filters are beneficial in the designing of effective and efficient multi-stage digital interpolators from the implementation perspective.

6.2 FUTURE SCOPE

In the aforementioned research work, we have proposed and simulated the CIC filters with compensation, the cubic Lagrange polynomial based Farrow filter structures, the conventional multi-stage filters and the half-band filters for interpolation in the evolving field of communication systems for the different wireless standards and engineering applications. The computational complexity in the hardware implementation is found to be in the trade-off relationship with efficiency of the presented interpolation structures. Therefore, the future work includes the designing and development of the nanotechnology based low cost, low power and highly efficient interpolation filter structures, which can support the fractional delay architectures. Moreover, the asynchronous signal reception as well as processing have generated an ample scope for further research using the advanced fractional delay Farrow filter structures.

LIST OF PUBLICATIONS

Refereed International Journals

1. R. Ratan, S. Sharma, and A. K. Kohli, "Performance Comparison of Quadrature Amplitude Modulation (QAM) Based Single and Double-stage Digital Interpolators," *International Journal of Electronics, Taylor and Francis*, vol. 100, no. 12, pp. 1724-1734, December 2013. (SCI Indexed; Impact Factor: 0.509).

   DOI: *http*://dx.doi.org/*10.1080/00207217.2013.769178*

2. R. Ratan, S. Sharma, and A. K. Kohli, "Cubic Lagrange Polynomial Based Designing of Efficient Interpolators," *International Journal of Electronics Letters, Taylor and Francis*, published online, 29th November 2013.

   DOI: *http://dx.doi.org/10.1080/21681724.2013.860574*

3. R. Ratan, S. Sharma, and A. K. Kohli, "Designing of Farrow Structure Based Interpolators on Nanometer based FPGA Technology," *Journal of Computational and Theoretical Nanoscience, American Scientific Publishers*, vol. 10, no. 9, pp. 2248-2259, September 2013. (SCI Indexed; Impact Factor: 0.673).

   DOI: *http://dx.doi.org/10.1166/jctn.2013.3194*

4. R. Ratan, S. Sharma, and A. K. Kohli, "Performance Comparison of QPSK Based Single and Double-stage Digital Interpolators," *Acta Physica Polonica-A*, vol. 123, no. 2, pp. 196-198, February 2013. (SCI Indexed; Impact Factor: 0.531).

   DOI: *http://dx.doi.org/10.12693/APhysPolA.123.196*

5. R. Ratan, S. Sharma, and A. K. Kohli, "Efficient Implementation of Multistage Digital Interpolators using Half Band Filters," *Acta Physica Polonica-A*, vol. 123, no. 2, pp. 199-201, February 2013. (SCI Indexed; Impact Factor: 0.531).

   DOI: *http://dx.doi.org/10.12693/APhysPolA.123.199*

6. R. Ratan, S. Sharma, and A. K. Kohli, “Effect of Compensation and Arbitrary Sampling on interpolators for Different Wireless Standards on FPGA Platform,” *Research Journal of Applied Sciences, Engineering and Technology*, Maxwell Scientific Organization, vol. 6 no. 4, pp. 609-621, June 2013 (Thomson Reuters Zoological Records)
7. R. Ratan, S. Sharma, and A. K. Kohli, “Performance Evaluation of Multirate Filters for Digital Down Converters,” *International Journal of the Physical Sciences*, vol. 6, no. 12, pp. 2807-2817, June 2011. (Formerly SCI Indexed when published; Impact Factor: 0.54).

DOI: *http://dx.doi.org/10.5897/IJPS11.362*

REFERENCES

[1] R. E. Crochiere and L. R. Rabiner, "Optimum FIR digital filter implementations for decimation, interpolation, and narrow-band filtering," *IEEE Trans. Acoust., Speech, Signal Process*., vol. 23, no. 5, pp. 444-456, October 1975.

[2] R.A. Meyer and C.S. Burrus, "Design and implementation of multirate digital filters," *IEEE Trans. Acous., Speech, Signal Process.,* vol. 24, no. 1, pp. 53-58, February 1976.

[3] M. Vetterli, "A theory of multirate filter banks," *IEEE Trans. Acoust., Speech, Signal Process.,* vol. 35, no. 5, pp. 356- 372, March 1987.

[4] P. Vaidyanathan, "Multirate digital filters, filter banks, polyphase networks, and applications: A tutorial," *Proc. IEEE*, vol. 78, no. 1, pp. 56-93, January 1990.

[5] P. P. Vaidyanathan, *Multirate Systems and Filter Banks.* Englewood Cliffs, New Jersey: Prentice Hall, 1993.

[6] E. B. Hogenauer, "An economical class of digital filters for decimation and interpolation," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 29, no. 2, pp. 155-162, April 1981.

[7] T. Hentschel and G. Fettweis, "Sample rate conversion for software radio," *IEEE Commun. Mag.,* vol. 38, no. 8, pp. 142-150, August 2000.

[8] A. A. W. Saud and G. Stuber, "Modified CIC filter for sample rate conversion in software radio systems," *IEEE Signal Process. Lett.,* vol. 10, no.5, pp. 152-154, May 2003.

[9] A. K. Z. Jiang and A. N. Willson Jr., "Application of filter sharpening to cascaded integrator-comb decimation filters," *IEEE Trans. Signal Process.,* vol. 45, no. 2, pp. 457-467, February 1997.

[10] G. Dolecek and S. Mitra, "A new two-stage sharpened comb decimator," *IEEE Trans. Circuits and Syst.-I,* vol. 52, no. 7, pp. 1414-1420, July 2005.

[11] P. Savvopoulos and T. Antonakopoulos, "An IF digital down converter for software radio DVB-S2 receivers," in *Proc. 14th IEEE Int. Conf. Elect., Circuits, Syst.*, December 2007, pp. 1043-1046.

[12] D. Babic, J. Vesma, and M. Renfors, "Decimation by irrational factor using CIC filter and linear interpolation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 2001, vol. 6, pp. 3677-3680.

[13] K. Chapman, *Building High Performance FIR Filters Using KCMs.* Xilinx Application Note, http://www.xilinx. com/appnotes/kcm_fir.pdf, 1996.

[14] T. Hentschel and G. Fettweis, "Continuous-time digital filter for sample rate conversion in reconfigurable radio terminals," in *Proc. European Wireless Conf.*, pp. 55-59, June 2000.

[15] A. A. W. Saud and G. Stuber, "Efficient wideband channelizer for software radio systems using modulated PR filter banks," *IEEE Trans. Signal Process.,* vol. 52 no. 10, pp. 2807-2820, October 2004.

[16] J. F. Kaiser and R.W. Hamming, "Sharpening the response of a symmetric non-recursive filters," *IEEE Trans. Acoust., Speech, Signal Process.,* vol. 25, no. 5, pp. 415-422, October 1977.

[17] H. J. Oh, S. Kim, G. Choi, and Y. Lee, "On the use of interpolated second order polynomials for efficient filter design in programmable down-conversion," *IEEE J. Selected Areas Commun.*, vol. 17, no. 4, pp. 551-560, April 1999.

[18] H. K. Yang and W. M. Snelgrove, "High speed polyphase CIC decimation filters," in *Proc. IEEE Int. Symp. Circuits, Syst.*, May 1996, vol. 2, pp. 229-232.

[19] H. Aboushady, Y. Dumonteix, M. Louerat, and H. Mehrez, "Efficient polyphase decomposition of comb decimation filters in ΣΔ analog-to-digital converters," *IEEE Trans. Circuits, Syst.-II,* vol. 48, no. 10, pp. 898-903, October 2001.

[20] J. Dunn, "Anti-alias and anti-image filtering: the benefits of 96 KHz sampling rate formats for those who can not hear above 20 KHz," in *Proc. 104th AES Conv.,* May 1998, pp. 1-8.

[21] P. S. Reddy, P. Satyanarayana, and M. N. S. Swamy, "Minimal multiplier realization of 1-D all-pass digital filters," *IEEE Trans. Circuits, Syst.*, vol. 37, no. 2, pp. 299-302, February 1990.

[22] S. K. Mitra and K. Hirano, "Digital all pass networks," *IEEE Trans. Circuits, Syst.,* vol. 21, no. 5, pp. 688-700, September 1974.

[23] K. Manivannan and C. Eswaran, "Minimal multiplier realization of 2-D all pass digital filters," *IEEE Trans. Circuits, Syst.*, vol. 35, no.4, pp. 480-482, April 1988.

[24] A. Y. Kwentus, Z. Jiang, and A.N. Willson Jr., "Application of filter sharpening to cascaded integrator-comb decimation filters," *IEEE Trans. Signal Process.*, vol. 45, no. 2, February 1997.

[25] C. W. Farrow, "A continuously variable digital delay element," in *proc. IEEE Int. Symp. Circuits, Syst.*, June 1988, vol. 3, pp. 2641-2645.

[26] J. Vesma, "Optimization and Applications of Polynomial-Based Interpolation Filters," Ph.D. dissertation, Tampere University of Technology, Tampere, Finland, 1999.

[27] T. A. Ramstad, "Digital methods for conversion between arbitrary sampling frequencies," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 32, no. 3, pp. 577-591, June 1984.

[28] S. Cucchi, F. Desinan, G. Parladori, and G. Sicuranza, "DSP implementation of arbitrary sampling frequency conversion for high quality sound application," in *proc. IEEE Int. Symp. Circuits, Syst.,* April 1991, vol. 5, pp. 3609-3612.

[29] C. C. Tseng, "Design of variable fractional delay FIR filter using differentiator bank," in *Proc. IEEE Int. Symp. Circuits, Syst., Phoenix,* May 2002, vol. 4, pp. 421-424.

[30] V. K. Jain and S. N. Gupta, "Performance of DPSK system in impulsive atmospheric radio noise with varying impulse rates and inter-symbol interference," *Electron. Lett.*, vol.19, no. 23, pp. 999-1000, November 1983.

[31] R. Dhiman, A. Kumar and R. Bahl, "Design of low power multi-DSP module for acoustic signal processing on remote platforms," in *Proc. Intl. Conf. Sonar-Sensors, Syst.,* December 2002, pp. 361-370.

[32] L. V. Subramaniam, B. S. Rajan, and R. Bahl, "Performance of 4 and 8-state TCM schemes with asymmetric 8-PSK in fading channels," *IEEE Trans. Veh. Tech.*, vol. 49, no. 1, pp. 211-219, January 2000.

[33] V. Gadre and R. K. Patney, "Vector multirate filtering and matrix filter banks," in *Proc. IEEE Int. Symp. Circuits, Syst. (ISCAS),* May 1992, vol. 3, pp.1360-1363.

[34] H. Johansson and P. Lowenborg, "Reconstruction of nonuniformly sampled bandlimited signals by means of digital fractional delay filters," *IEEE Trans. Signal Process.*, vol. 50, no. 11, pp. 2757–2767, November 2002.

[35] U. Nithirochananont, S. Chivapreecha, and K. Dejhan, "An FPGA implementation of variable fractional delay filter," in *Proc. IEEE 5th Int. Colloq Signal Process. App.*, March 2009, pp. 104-107.

[36] D. Babic and M. Renfors, "Power efficient structures for conversion between arbitrary sampling rates," *IEEE Signal Process. Lett.*, vol. 12, no. 1, pp. 1-4, January 2005.

[37] A. Franck and K. Brandenburg, "A closed form description for the continuous frequency response of Lagrange interpolators," *IEEE Signal Process. Lett.*, vol. 16, no. 7, July 2009.

[38] C. H. Dick, "Implementing area optimized narrow band FIR filters using Xilinx FPGAs," in *Proc. SPIE's Photonics East'98 Config. Computing Tech., App., Boston, MA, USA*, May 1998. pp. 227-238.

[39] C. H. Dick and F. Harris, "FPGA interpolators using polynomial filters," in *Proc. 8th Int. Conf. Signal Process. App., Tech.*, June 1998, vol. 1, pp. 684-688.

[40] V. Valimaki, "A new filter implementation strategy for Lagrange interpolation," in *Proc. IEEE Int. Symp. Circuits, Syst. (ISCAS-95)*, May 1995, vol. 1, pp. 361-364.

[41] G. J. Cook, Y. H. Leung, and Y. Liu, "On the design of variable fractional delay filters with Laguefcre and Kautz filters," in *Proc. IEEE Int. conf. Acoust., Speech, Signal Process. (ICASSP '03)*, April 2003, vol. 6, pp. 281-284.

[42] T. Deng, "Discretization-free design of variable fractional delay FIR digital filters," *IEEE Trans. Circuits, Syst.*, vol. 48, no. 6, pp. 637-644, June 2001.

[43] L. S. H. Ngia, "Separable nonlinear least-squares methods for efficient off-line and on-line modeling of systems using Kautz and Laguerre filters," *IEEE Trans. Circuits, Syst.*, vol. 48, no. 6, pp. 562-579, June 2001.

[44] G. Watkins, "Optimal Farrow coefficients for symbol timing recovery," *IEEE Commun. Lett.*, vol. 5, no. 9, pp. 381-383, September 2001.

[45] S. Usui and I. Amidror, "Digital low-pass differentiation for biological signal processing," *IEEE Trans. Biomed. Eng.*, vol. 29, no. 10, pp. 686-693, October 1982.

[46] T. W. Parks and J. H. McClellan, "Chebyshev approximation for non-recursive digital filters with linear phase," *IEEE Trans. Circuit Theory*, vol. 19, no. 3, pp. 189-194, March 1972.

[47] S. C. Pei and J. J. Shyu, "Eigenfilter design of higher-order digital differentiators," *IEEE Trans. Acoust. Speech, Signal Process.*, vol. 37, pp. 505-511, April 1989.

[48] B. Carlsson, "Maximum flat digital differentiator," *IEEE Electron. Lett.*, vol. 27, no. 8, pp. 675-677, April 1991.

[49] S. Sunder and V. Ramachandran, "Design of equi-ripple non recursive digital differentiators and Hilbert transformers using a weighted least squares technique," *IEEE Trans. Signal Process.*, vol. 42, no. 9, pp. 2504-2509, September 1994.

[50] C. C. Tseng, "Stable IIR digital differentiator design using iterative quadratic programming approach," *Signal Process.*, vol. 80, no. 5, pp. 857-866, May 2000.

[51] I. R. Khan and R. Ohba, "New design of full band differentiators based on Taylor series," *IEEE Vis. Image Signal Process.*, vol. 146, no. 4, pp. 185-189, August 1999.

[52] I. R. Khan and R. Ohba, "Taylor series based two-dimensional digital differentiators," *Proc. IEEE Vis. Image Signal Process.*, vol. 149, no. 4, pp. 231-236, August 2002.

[53] W. S. Lu and T. B. Deng, "An improved weighted least-squares design for variable fractional delay FIR filters," *IEEE Trans. Circuits, Syst. II, Analog Digit. Signal Process.*, vol. 46, no. 8, pp. 1035-1040, August 1999.

[54] T. B. Deng and W. S. Lu, "Weighted least-squares method for designing variable fractional delay 2-D FIR filters," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 47, no. 2, pp. 114-124, February 2000.

[55] C. C. Tseng, "Design of 1-D and 2-D variable fractional delay all-pass filters using weighted least-squares method," *IEEE Trans. Circuits, Syst. I, Fundam. Theory Appl*., vol. 49, no. 10, pp. 1413-1422, October 2002.

[56] C. C. Tseng, "Eigen filter approach for the design of variable fractional delay FIR and allpass filters," *Proc. IEEE Vis. Image Signal Process*., vol. 149, no. 5, pp. 297-303, October 2002.

[57] T. B. Deng and Y. Nakagawa, "SVD-based design and new structures for variable fractional-delay digital filters," *IEEE Trans. Signal Process.*, vol. 52, no. 9, pp. 2513-2527, September 2004.

[58] C. C. Tseng, "Digital differentiator design using fractional delay filter and limit computation," *IEEE Trans. Circuits, Systems*, vol. 52, no. 10, pp. 2248-2259, October 2005.

[59] M. Satish and S. Gupta, "Design and application of a new multi-scale multidirectional non sub-sampled filter bank," in *Proc. 8th Int. Conf. Signal Image Tech., Internet Based syst.*, November 2012, pp. 64-71.

[60] F. Harris, "Performance and design of Farrow filter used for arbitrary re-sampling," in *Proc. 13th Int. Conf. Digit. Signal Process.*, July 1997, vol. 2, pp. 595-599.

[61] J. Selva, "Interpolation of bounded band-limited signals and applications," *IEEE trans. Signal Process.*, vol. 54, no. 11, pp. 4244-4260, November 2006.

[62] S. Bose, V. Khaitan, and A. Chaturvedi, "A low-cost algorithm to find the minimum sampling frequency for multiple bandpass signals," *IEEE Signal Process. Lett.*, vol. 15, pp. 877-880, December 2008.

[63] B. Patil, G. Sharma, P. Patwardhan, and V. Gadre, "A generalized approach for finite precision 5/3 Filter Design," in *Proc. National Commun. Conf. (NCC)*, January 2007, pp 1-4.

[64] V. Lehtinen, D. Babic, and M. Renfors, "Comparison of continuous and discrete-time modeling of polynomial-based interpolation filters," in *Proc. 6th Nordic Signal Process. Symp. (NORSIG 2004)*, June 2004, pp. 49-52.

[65] R. Lagadec and H. Kunz, "A universal digital sampling frequency converter for digital audio," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, April 1981, vol. 6, pp. 595-598.

[66] R Lagadec, D. Pelloni, and D. Weiss, "A 2-channel 16-bit digital sampling frequency converter for professional digital audio," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 1982, vol. 7, pp. 93-97.

[67] J. W. Adam, R.W. Bayma, and J.E. Nelson, "Digital filter design for generalized interpolation," in *Proc. IEEE Int. Symp. Circuits, Syst. (ISCAS'89)*, May 1989, vol. 2, pp. 1299-1302.

[68] G. R. Goslin, "A guide to using field programmable gate arrays for application specific DSP performance," in *Proc. (SPIE 2914) High-Speed Computing, Digit. Signal Process., Filtering Using Reconfig. Logic,* October 1996. http://www.xilinx. com/appnotes/dspguide.pdf.

[69] R. E. Crochiere and L. R. Rabiner, "Interpolation and decimation of digital signals - A tutorial review," *Proc. IEEE*, vol. 69, no. 3, pp. 300-331, March 1981.

[70] R. E. Crochiere, L. R. Rabiner, “Further considerations in the design of decimators and interpolators,” *IEEE Trans. Acoust., Speech, Signal Process.,* vol. 24, no. 4, pp. 296-311, August 1976.

[71] C. H. Dick and F. Harris, “High performance FPGA filters using sigma-delta modulation encoding,” in *Proc. Int. Conf. Acoust. Speech, Signal Process. (ICASSP),* March 1999, vol. 4, pp. 2123-2126.

[72] I. Lokken, “The ups and downs of arbitrary sample rate conversion, an application note,” *www.iet.ntnu.no/courses/fe8114/slides/upsanddownsofasrc.pdf,* December 2005.

[73] L. Erup, F. M. Gardner, and R. A. Harris, “Interpolation in digital modems-part II: implementation and performance,” *IEEE Trans. Commun.*, vol. 41, no. 6, pp. 998-1008, June 1993.

[74] A. S. H. Ghadam, D. Babic, V. Lehtinen, M. Renfors, “Implementation of Farrow structure based interpolators with sub-filters of odd length,” in *Proc. IEEE Int. Symp. Circuits, Syst.*, May 2004, vol. 3, pp. 581-584.

[75] J. Vesma and T. Saramaki, “Polynomial-based interpolation filters - part I: filter synthesis,” *Springer link Circuits Syst., Signal Process. Lett.*, vol. 26, no. 2, pp. 115-146, April 2007.

[76] C. Candan, “An efficient filtering structure for Lagrange interpolation,” *IEEE Signal Process. Lett.*, vol. 14, no. 1, pp. 17-19, January 2001.

[77] S. Samadi, M. O. Ahmad, and M. N. S. Swamy, “Results on maximally flat fractional-delay systems,” *IEEE Trans. Circuits Syst. I*, vol. 51, no. 11, pp. 2271-2286, November 2004.

[78] D. Luengo, C. Pantaleon, J. Ibanez, and I. Santamaria, "Design of simultaneous sampling systems based on fractional delay Lagrange filters," *IEEE Trans. Circuits Syst. II*, vol. 47, no. 5, pp. 482-485, May 2000.

[79] A. G. Dempster and N. P. Murphy, "Efficient interpolators and filter banks using multiplier blocks," *IEEE Trans. Signal Process.*, vol. 48, no. 1, pp. 257-261, January 2000.

[80] F. Harris, *Multirate Signal Processing for Communication Systems*. 1st ed. US: Prentice Hall, 2004.

[81] F. Harris, *Multirate Signal Processing for Communication Systems.* 2nd ed., New Delhi, India: Pearson Education, 2009.

[82] S.H. Yoon and J.W. Chong, C.H. Lin, "An area optimization method for digital filter design," *Electronics and Telecommunication Research Institute (ETRI) J.*, vol. 26, no. 6, pp. 545-554, December 2004.

[83] J. Vesma and T. Saramaki, "Design and properties of polynomial-based fractional delay filters," in *Proc. IEEE Int. Symp. Circuits, Systems*, May 2000, pp. 104-107.

[84] J. Vesma and T. Saramaki, "Optimization and efficient implementation of FIR filters with adjustable fractional delay," in *Proc. IEEE Int. Symp. Circuits, Syst.*, June 1997, vol. 4, 2256-2259.

[85] P. Vaidyanathan, "Generalizations of the sampling theorem: seven decades after Nyquist," *IEEE Trans. Circuits and Syst. I: Fundamental Theory and Applications*, vol. 48, no. 9, pp. 1094-1109, September 2001.

[86] B. D. Patil, P. G. Patwardhan, and V. M. Gadre, "Filter bank design using a factorization of parametrized half-band filters," in *Proc. 14th Nat. Conf. Commun. (NCC)*, February 2008, pp. 404-408.

[87] D. Babic, “Piecewise polynomial approximation based on Taylor series with efficient realization using Farrow structure,” in *Proc. IEEE int. conf. TELSIKS Serbia*, October 2009, pp. 241-244.

[88] T. A. Ramstad, “Fractional rate decimator and interpolator design,” in *Proc. Int. conf. EUSIPCO*, September 1998, pp. 49-52.

[89] T. A. Ramstad, “Flexible and efficient interpolator design,” in *Proc. IEEE Norwegian Signal Process. Symp.(NORSIG’ 99)*, May 1999, pp. 81-85.

[90] M. Kiessling and A. A. Mujtaba, “Software radio architecture for multi-channel digital up-conversion and down-conversion using generalized polyphase filter banks with frequency offset correction,” in *Proc. IEEE Personal Indoor Mobile Radio Conf.*, September 2002, vol. 10, pp. 105-109.

[91] N. B. Ameur and M. Loulou, “Design of efficient digital interpolation filters and sigma-delta modulator for an audio DAC,” in *Proc. IEEE 3rd int. conf. Design, Tech. Integrated Syst. in Nanoscale era*, March 2008, pp. 1-7.

[92] S. S. Lokesh, A. Kumar, and M. Agrawal, “Structure of an optimum linear precoder and its application to ML equalizer,” *IEEE Trans. Signal Process.*, vol. 56, no. 8, pp. 3690-3701, August 2008.

[93] A. Abidi, “The path to the software defined radio receivers,” *IEEE Trans. Circuits, Syst.*, vol. 42, no. 5, pp. 954-966, May 2007.

[94] H. M. Seo, C. G. Woo, and P. Choi, “Relationship between ADC performance and requirements of digital-IF receiver for WCDMA base-station,” *IEEE Trans. Veh. Tech.*, vol. 52, no. 5, pp. 1398-1408, Sepember 2003.

[95] V. Arkesteijn, E. Klumperink, and B. Nauta, “Jitter requirements of the sampling clock in software radio receivers,” *IEEE Trans. Circuits, Syst. II: Express Briefs*, vol. 53, no. 2, pp. 90-94, February 2006.

[96] F. Dengwei, N. Alan, and J. Willson, “Design of an improved interpolation filter using a trigonometric polynomial,” in *Proc. IEEE Int. Symp. Circuits, Syst.*, July 1999, vol. 4, pp. 363-366.

[97] F. Francesconi, G. Lazzari, V. Liberali, F. Maloberti, and G. torelli, “Multiplier-free Lagrange interpolators for oversampled D/A converters,” in *Proc. IEEE Int. Symp. Circuits, Syst.*, May 1993, vol. 1, pp. 219-222.

[98] H. Choi and D. C. Munson, “Analysis and design of minimax-optimal interpolators,” *IEEE Trans. Signal Process.*, vol. 46, no. 6, pp. 1571-1579, June 1998.

[99] W. H. Yim, “Distortion analysis for multiplierless sampling rate conversion using linear transfer functions,” *IEEE Signal Process. Lett.*, vol. 8, no. 5, pp. 143-144, May 2001.

[100] D. Babic, V. Lehtineni, and M. Renfors, “Discrete-time modeling of polynomial-based interpolation filters in rational sampling rate conversion,” in *Proc. Int. Symp. Circuits, Syst. (ISCAS 2003),* May 2003, vol. 4, pp. 321-324.

[101] D. Babic, “Optimization of polynomial-based interpolation filters using discrete-time model,” in *Proc. 15th Telecom. forum (TELFOR),* November 2007, pp. 20-22.

[102] J. G. Proakis and M. Salehi, *Digital Communication.* 5th ed. New York, USA: McGraw-Hill, 2007.

[103] J. Vesnia, V. Tuukkanen, and M. Rcnfors, “Maximurn likelihood feed-forward symbol timing recovery based on efficient digital interpolation techniques,” in *Proc. IEEE Nordic Signal process. Symp.,* September 1996, pp. 183-186.

[104] J. Vesma, V. Tuukkanen, R. Hamila, and M. Renfors, "Block-based feedforward maximum likelihood symbol timing recovery technique," *in Proc. Commun. Syst., Digit. Signal Process.,* April 1998, pp. 92-95.

[105] R. Hamila, J. Vesma, H. Vuolle, and M. Renfors, "Effect of frequency offset on carrier, phase and symbol timing recovery in digital receivers," in *Proc. IEEE Int. Symp. Signals, Syst., Elect.*, October 1998, pp. 247-252.

[106] R. E. Crochiere and L. R. Rabiner, *Multirate Digital Signal Processing*. Englewood Cliffs, New Jersey: Prentice-Hall, 1983.

[107] R. E. Crochiere, S.A. Webber, and J. L. Flanagan, "Digital coding of speech in subbands," *Bell Syst. Tech. J.,* vol. 55 no. 8, pp.1069-1085, October 1976.

[108] S. Mitra, *Digital Signal Processing: A Computer Based Approach*. New Delhi, India: Prentice Hall, 2000.

[109] K. Yeung and S. C. Chan, "The design and multiplier less realization of software radio receivers with reduced system delay," *IEEE Trans. Circuits, Syst.,* vol. 51, no. 12, pp. 2444-2459, December 2004.

[110] P. P. Vaidyanathan and T. Q. Nguyen, "A trick for the design of FIR half-band FIR filters," *IEEE Trans. Circuits, Syst.*, vol. 34, no. 3, pp. 297-300, March 1987.

[111] C. J. Chou, M. Satish, and B. E. Joseph, "FPGA implementation of digital filters," in *Proc. Int. Conf. ICSPAT*, August 1993, pp. 1-9.

[112] M. G. Bellanger, J. L. Daguet, and G. P. Lepagnol, "Interpolation, extrapolation, and reduction of computation speed in digital filters," *IEEE Trans. Acoust., Speech, Signal Process.,* vol. 22, no. 4, pp. 231-235, August 1974.

[113] M. G. Bellanger, "Computation rate and storage estimation in multirate digital filtering with half-band filter," *IEEE Trans. Acoust., Speech, Signal Process,* vol. 25, no. 4, pp. 344-346, August 1977.

[114] D. J. Goodman, "Nine digital filters for decimation and interpolation," *IEEE Trans. Acoust., Signal Process.*, vol. 25, no. 2, pp. 121-126, April 1977.

[115] K. Lin, K. Zhao, E. Chui, A. Krone, and J. Nohrden, "Digital filters for high performance audio delta-sigma analog to digital and digital to analog conversions," in *Proc. 3rd Int. Conf. Signal Process.*, October 1996, vol. 1, pp. 59-63.

[116] I. Fujimori and T. Sugimoto, "A 1.5V, 4.1mW dual channel audio delta-sigma D/A converter," in *Proc. IEEE Int. Conf. Sol. Sta. Circ.*, February 1998. pp. 60-61.

[117] X. Huang, Y. Han, and L. Chen, "The design and FPGA verification of a general structure, area-optimised interpolation filter used in delta sigma DAC," in *Proc. 8th Int. Conf. Solid State, Integrated circuit Tech.*, October 2006, pp. 2111-2113.

[118] A. Mukhtar, H. Jamal, and U. Farooq, "An area efficient interpolation filter for digital audio applications," *IEEE Trans. Consumer Electron.*, vol. 55, no. 2, pp. 768-772, May 2009.

[119] H. Wang, C. Chan, and C. Choy, "CMOS high speed interpolators based on parallel architecture," *IEEE Trans. Consumer Electron.,* vol. 46, no. 2, pp. 326-329, May 2000.

[120] E. M. Ashmila, S. S. Dlay, and O. R. Hinton, "Adder methodology and design using probabilistic multiple carry estimates," *IEEE Proc. Computer, Digit. Tech.*, vol. 152, no. 6, pp. 697-703, November 2005.

[121] A. Kumar, R. S. Anand, and G. K. Singh, "Near perfect reconstruction quadrature mirror filter," *Int. J. of Computer Science and Eng.*, vol.2, no. 3, pp. 121-124, March 2008.

[122] M. J. Licea, S. V. Beltran, and J. L. Bonilla, "Experimental performance analysis for 2G/3G cellular networks based on mobile terminals," *J. Institution Eng.*, vol. 8, no. 1-2, pp. 25-38, July 2011.

[123] A. Gupta, S. D. Joshi, and S. Prasad, "A new approach for estimation of statistically matched wavelet," *IEEE Trans. Signal Process.*, vol. 53, no. 5, pp. 1778-1793, May 2005.

[124] R. R. Mahmud, M. A. G. Khan, and S. M. A. Razzak, "Design of a duobinary encoder and decoder circuits for communication systems," in *Proc. 6th IEEE Int. Conf. Electri., Computer Eng. (ICECE-2010)*, December 2010, pp. 49-52.

[125] M. Karimi, V. Tahani, and S. Gazor, "Matrix interpolation: some control applications" *Int. J. Control*, vol. 72, no. 2, pp. 174-192, February 1999.

[126] C. S. Anand and J. S. Sahambi, "MRI denoising using bilateral filter in redundant wavelet domain," in *Proc. IEEE region 10 conf. (TENCON 2008)*, November 2008, pp. 1-6.

[127] R. E. Crochiere and L. R. Rabiner, "Recent developments in the design and implementation of digital decimators, interpolators and narrow band filters," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP '76)*, April 1976, vol. 1, pp. 292-295.

[128] D. B. Turek, "Design of efficient digital interpolation filters for integer up-sampling," M. E. Thesis, Massachusetts Institute of Technology, June 2004.

[129] M. Feldbauer, K.W Kepesi, and E. Rank, "*Multirate Signal Processing v1.3.3.* Signal Processing and Speech Communication Laboratory Handout, 2006.

[130] C. H. Lu and S. C. Gupta, "Optimal design of multi-rate digital filters with application to interpolation," *IEEE Trans. Signal Process.*, vol. 26, no. 3, pp. 160-165, March 1979.

[131] R. W. Schaffer, L. R. Rabiner, "A digital signal processing approach to Interpolation," *Proc. IEEE,* vol. 61, no. 6, 692-702, June 1973.

[132] L. R. Rabiner, J. H. McClellan, and T. W. Parks, "FIR digital filter design techniques using weighted Chebyshev approximation," *Proc. IEEE*, vol. 63, no. 4, pp. 595-610, April 1975.

[133] R. A. Losada, "*Digital Filters with MATLAB.* The MathWorks Inc., 2009.

[134] M. G. Bellanger, G. Bonnerot, and M. Coudreuse, "Digital filtering with polyphase network: application to sample rate alteration and filter banks," *IEEE Trans. Acoust., Speech, Signal Process.,* vol. 24, no. 2, pp. 109-114, April 1976.

[135] T. W. Parks and D. P. Kolba, "Interpolation minimizing maximum normalized error for band-limited signals," *IEEE Trans. Acoust., Speech, SignaI Process.*, vol. 26, no. 2, pp. 381-384, August 1976.

[136] A. D. Polydoros and E. N. Protonotarios, "Digital interpolation of stochastic signals," *IEEE Trans. Circuits, Syst.,* vol. 26, no. 11, pp. 916-922, November 1979.

[137] M. P. Donadio, "CIC Filter Introduction," *Iowegian Publication,* 2000 (http://www.mikrocontroller.net/attachment/51932/cic2.pdf)

[138] T. Saramaki and T. Ritoniemi, "An efficient approach for conversion between arbitrary sampling frequencies," in *Proc. IEEE Int. Symp. Circuits, Syst.*, May 1996, vol. 2, pp. 285-288.

[139] U. Meyer-Bease, *Digital Signal Processing with Field Programmable Gate Arrays.* 3rd ed. London, UK: Springer, 2007.

[140] M. Ljiljana, *Multirate Filtering for Digital Signal Processing: MATLAB Applications.* Hershey, PA: Information Science Reference, 2009.

[141] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms and Applications.* 4th ed. New Delhi: PHI Publications Inc., 2006.

[142] Altera White Paper, WP-01097, "Simplifying simultaneous multimode RRH design," pp. 1-13, 2009. Available at: http://www.altera.com/literature/wp/wp-01097-arria-ii-gx-multimode-rrh.pdf.

[143] J. Vesma, "A frequency-domain approach to polynomial based interpolation and the Farrow structure," *IEEE trans. on Circuits, Syst. II: Anal. Digit. Signal Process.*, vol. 47, no. 3, pp. 206-209, March 2000.

[144] F. Harris, "Performance and design considerations of the Farrow filter when used for arbitrary re-sampling of sampled time series," in *Proc. IEEE 31st Asilomar Conf. Signals, Systems, Comput.*, November 1997, vol. 2, pp. 1745-1749.

[145] A. S. H. Ghadam, M. Renfors, "Farrow structure interpolators based on even order shaped Lagrange polynomials," in *Proc. IEEE 3rd Int. Symp. Image, Signal Process., Analysis*, September 2004, vol. 2, pp. 745-748.

[146] Y. Zhuan, "Linear phase Lagrange interpolation filter using odd number of base-points," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.,* April 2003, vol. 6, pp. 237-240.

[147] V. Valimaki, "Discrete-Time Modeling of Acoustic Tubes Using Fractional Delay Filters," Ph. D. thesis, Helsinki University of Technology, 1995.

[148] C. Moler, *Numerical Computing with MATLAB.*1st ed. USA: Siam publications, 2004.

[149] Altera Application note, "DSP Builder Handbook vol. 3: DSP Builder Advanced Blockset," *Available at:*

www.altera.com/literature/hb/dspb/hb_dspb_adv.pdf, 2011.

[150] L. Milic, *Multirate Filtering for digital Signal Processing: MATLAB Applications.* London, United Kingdom: Information Science Reference, 2009.

[151] S. Sharma, *Signals and Systems.* 4th ed. New Delhi, India: S.K. Kataria and Sons, 2005.

[152] S. Salivahanan, A. Vallavaraj, C. Gnanapriya, *Digital Signal Processing.* 1st ed. New Delhi: Tata McGraw-Hill Publications Company Ltd., 2000.

[153] T. Saramaki, *Handbook for Digital Signal Processing.* New York, USA: John Wiley Interscience, 1993.

[154] T. Xiaoyi, M. S. Alouini, and A. Goldsmith, "Effect of channel estimation error on M-QAM BER performance in Rayleigh fading," in *Proc. IEEE 49th Veh. Tech. Conf.*, July 1999, vol. 2, pp. 1111-1115.

[155] L. R. Rabiner and B. Gold, *Theory and Application of Digital Signal Process.* Englewood Cliffs, NJ: Prentice-Hall, 1975.

[156] F. Mintzer, "On half-band, third-band, and nth-band FIR filters and their design," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 30 no. 5, pp.734-738, October 1982.

[157] T. Saramaki, and M. Renfors, "Nth band filter design," in *Proc. 9th European Signal Process. Conf., EUSIPCO,* September 1998, pp. 1943-1947.

[158] N. J. Fliege, *Multirate Digital Signal Processing.* New York, USA: John Wiley and Sons, 1994.