

# **Area Efficient Reconfigurable Architectures for Sample Rate Conversion in SDR Receivers**

*Submitted in partial fulfilment of the requirements*

*for the award of the degree of*

**Doctor of Philosophy**

by

**Ashok Agarwal**

(Roll No: 701143)

Under the supervision of

**Dr. B. Lakshmi**



**Department of Electronics & Communication Engineering**

**National Institute of Technology Warangal**

**Telangana, India - 506004**

**2019**

---

Dedicated  
To  
My Family,  
Teachers & Friends

## Approval Sheet

This thesis entitled **Area Efficient Reconfigurable Architectures for Sample Rate Conversion in SDR Receivers** by **Ashok Agarwal** is approved for the degree of **Doctor of Philosophy**.

### **Examiners**

---

---

### **Research Supervisor**

---

**Dr. B. Lakshmi**

Department of ECE

NIT Warangal, India-506004

### **Chairman & Head**

---

**Dr. N. Bheema Rao**

Department of ECE

NIT Warangal, India-506004

Place:

Date:

## Declaration

This is to certify that the work presented in this thesis entitled **Area Efficient Reconfigurable Architectures for Sample Rate Conversion in SDR Receivers** is a bonafied work done by me under the supervision of **Dr. B. Lakshmi** and was not submitted elsewhere for the award of any degree.

I declare that this written submission represents my own ideas and even considered others ideas which are adequately cited and further referenced the original sources. I understand that any violation of the above will cause disciplinary action by the institute and can also evoke panel action from the sources or from whom proper permission has not been taken when needed. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea or data or fact or source in my submission.

Place:

Date:

Ashok Agarwal  
Research Scholar  
Roll No.: 701143

# NATIONAL INSTITUTE OF TECHNOLOGY

WARANGAL, INDIA-506004

Department of Electronics & Communication Engineering



## CERTIFICATE

This is to certify that the thesis work entitled **Area Efficient Reconfigurable Architectures for Sample Rate Conversion in SDR Receivers** is a bonafide record of work carried out by **Ashok Agarwal** submitted to the faculty of **Electronics & Communication Engineering** department, in partial fulfilment of the requirements for the award of the degree of **Doctor of Philosophy in Electronics and Communication Engineering, National Institute of Technology Warangal, India-506004**. The contributions embodied in this thesis have not been submitted to any other university or institute for the award of any degree.

Dr. B. Lakshmi

Research Supervisor

Place:

Associate Professor

Date:

Department of ECE

NIT Warangal, India-506 004.

## **Acknowledgements**

At the outset, I take immense pleasure to convey my sincere gratitude to my supervisor Dr.B.Lakshmi, Associate Professor, Department of Electronics & Communication Engineering, National Institute of Technology, Warangal, for her perpetual encouragement and supervision. Her steady influence throughout my Ph.D. career has oriented me in a proper direction and supported me with promptness and care.

I thank all the faculty and non-teaching staff of Dept. of ECE at NIT Warangal who helped me during the course. I am also grateful to Prof. N. Bheema Rao, Head, Department of ECE, for his invaluable assistance and suggestions that he shared during my research tenure.

I take this privilege to thank all my Doctoral Scrutiny Committee members, Prof. C.B.Rama Rao, Professor of Department of ECE, and Prof. G. Radhakrishnamacharya, Professor of Mathematics Department, for their detailed review, constructive suggestions and excellent advice during the progress of this research work.

I thank my nation India, for giving me the opportunity to carry out my research work at the NIT Warangal. A special thanks to MHRD for its financial support.

I also extend my heartfelt appreciation to all my colleague scholars, friends and well-wishers who helped to write my thesis with their support. Finally I would like to acknowledge my biggest debt, to my parents and my sister for their constant support.

**Ashok Agarwal**

## Abstract

Due to recent advancements in wireless technology, there is a tremendous growth in the wireless industry and huge demand for high data rate applications. Over the past two decades, the focus of researchers has been to investigate new radio communication standards, such that the data rates as high as possible can be supported. This led to the changes in the radio communication standards at a faster rate. Radio communication systems when implemented as a hardware radio, the hardware becomes obsolete or needs to be redesigned with the changing radio standards. Solution to address this problem is Software Defined Radio (SDR).

A radio transceiver consists of Baseband (BB) processing stage, Intermediate Frequency (IF) processing stage and Radio Frequency (RF) processing stage. In an ideal SDR, all the three stages are implemented in software by placing data converters immediately after the antenna. Due to practical limitations of data converters, a practical SDR architecture is obtained by placing data converters at IF stage. In our work, we have focused on the design of IF stage of SDR receiver.

At the IF stage of SDR receiver, if the wideband signal is digitized at Nyquist rate the narrow band radio channels gets oversampled due to the phenomenon of pass band sampling. The signal processing carried out at this sampling rate leads to high power dissipation in the further stages of SDR receiver. Hence, a sample rate converter (SRC) is required to decimate the sample rate as per the specifications of radio standard into consideration. In SDR receivers, it is required to achieve SRC by programmable integer rate or programmable fractional rate or both. Integer rate SRC achieves SRC by large rates, while, the fractional rate SRC is required to make the signal sample rate power of two multiple of symbol rate of the radio standard. Hence, it is required to design a sample rate converter with reconfigurable SRC factors and tunable spectral characteristics while

---

## Abstract

---

achieving minimum reconfiguration overhead and low hardware complexity.

Coefficient-less cascaded-integrator-comb (CIC) filters, a special class of filters proposed in the literature are suitable for achieving SRC by large integer factors as required in SDR receivers. These filters require less power consumption and can be reconfigured flexibly due to their coefficient-less architectures. CIC filters introduce gain droop in the pass band of interest and cannot achieve SRC by fractional rates. Hence, it is required to cascade a gain droop compensation and fractional rate interpolation filter.

In our work, we have considered four radio standards, WiMAX, WCDMA, CDMA2000 and GSM900 for multi-standard radio implementation with an IF frequency of 80MHz. We have employed CIC filters followed by compensation and interpolation filters designed based on discrete compensation and interpolation method, joint compensation and interpolation method and simulated for their frequency response in MATLAB. Compensation filters designed for one radio standard employing these methods require offline computations to support new radio communication standard.

In the present work, we make an attempt to eliminate the need of offline computations. We propose to apply singular value decomposition based variable digital filter (SVD-VDF) technique to design a joint compensation and interpolation filter. The SVD-VDF consists of fixed coefficient sub-filters and multi-dimensional polynomials in terms of spectral parameters to reconfigure the spectral characteristics. Further, the hardware complexity of proposed SVD-VDF based joint compensation and interpolation filter is reduced by proposing multiplier-less distributed arithmetic (DA) architecture for implementation of sub-filters in the SVD-VDF filter.

The functionality of the proposed reconfigurable SRC filter is verified by simulating the VHDL netlist and implementing on Kintex-7 XC7k325t-2ffg900 FPGA using Xilinx Vivado 2014.2 software. The performance of the proposed DA based architectures is computed by synthesizing using Synopsis Design Compiler with TSMC CMOS 90nm technology library. The performance comparison of these proposed DA-based architectures with the DA-based architectures available in the literature shows improvement in hardware complexity and power consumption.

---

# Contents

<b>Declaration</b>	iii
<b>Acknowledgements</b>	v
<b>Abstract</b>	vi
<b>List of Figures</b>	xiii
<b>List of Tables</b>	xvi
<b>List of Abbreviations</b>	xviii
<b>1 Introduction</b>	1
1.1 Motivation . . . . .	1
1.2 Research Objectives . . . . .	2
1.3 Thesis Contributions . . . . .	3
1.4 Thesis Organization . . . . .	6
1.5 Conclusions . . . . .	7
<b>2 Software Defined Radio</b>	8
2.1 Radio Communication Systems . . . . .	8
2.2 Ideal SDR . . . . .	13

---

2.3	Digital Front-End . . . . .	15
2.3.1	SDR Transmitter . . . . .	16
2.3.2	SDR Receiver . . . . .	18
2.4	Conclusions . . . . .	20
<b>3</b>	<b>Digital Front-End Stage of SDR Receivers</b>	<b>21</b>
3.1	Channelization . . . . .	21
3.1.1	Per-Channel Approach . . . . .	22
3.1.2	Discrete Fourier transform Filter Bank Approach . . . . .	22
3.1.3	Pipelined Frequency Transform Approach . . . . .	24
3.1.4	Frequency Response Masking Filter Bank approach . . . . .	25
3.1.5	Coefficient Decimation Filter Bank Approach . . . . .	27
3.1.6	Comparison of various channelization approaches . . . . .	28
3.2	Digital Down Conversion-Sample Rate Conversion . . . . .	29
3.2.1	Quadrature Mixers . . . . .	30
3.2.2	Programmable Sample Rate Conversion . . . . .	31
3.3	Multi-Rate SRC filters . . . . .	33
3.3.1	FIR filter based SRC structures . . . . .	34
3.3.2	Coefficient-less/Fixed Coefficient SRC structures . . . . .	37
3.4	Gain droop compensation and Interpolation filters . . . . .	40
3.4.1	Discrete Compensation and Interpolation filter . . . . .	40
3.4.2	Joint compensation and interpolation filter . . . . .	44
3.5	Conclusions . . . . .	44
<b>4</b>	<b>Sample Rate Conversion Filter for SDR Receivers</b>	<b>46</b>

---

4.1	Introduction . . . . .	46
4.2	Reconfigurable Architecture for Sample Rate Conversion Filter . . . . .	47
4.2.1	Design specifications . . . . .	47
4.2.2	Programmable SRC Architecture for SDR receivers . . . . .	48
4.3	Integer Rate Sample Rate Conversion . . . . .	50
4.3.1	CIC Filter Design . . . . .	50
4.3.2	Multi-stage CIC Filter Design . . . . .	52
4.3.3	FPGA Implementation . . . . .	54
4.3.4	Frequency Response . . . . .	54
4.3.5	Summary . . . . .	56
4.4	CIC Compensation and Interpolation filter . . . . .	56
4.4.1	Discrete Gain Compensation and Interpolation filter . . . . .	57
4.4.2	Proposed Joint Compensation and Interpolation Filter . . . . .	60
4.4.3	Architectures . . . . .	65
4.5	Implementation Results . . . . .	67
4.6	Conclusions . . . . .	68
<b>5</b>	<b>SVD based Reconfigurable SRC Filter for SDR Receivers</b>	<b>70</b>
5.1	Introduction . . . . .	70
5.2	Singular Value Decomposition (SVD) Algorithm . . . . .	71
5.3	SVD-VDF Design . . . . .	72
5.4	Proposed SVD-VDF based Joint Compensation and Interpolation Filter . .	75
5.5	Performance Comparison . . . . .	81
5.5.1	Frequency Response . . . . .	81
5.5.2	Hardware Complexity . . . . .	83

---

---

5.6 Conclusions . . . . .	86
<b>6 Area Efficient Architecture for Reconfigurable SRC Filters in SDR Receivers</b>	<b>87</b>
6.1 Introduction . . . . .	87
6.2 Distributed Arithmetic Algorithm based FIR Filters . . . . .	88
6.3 Distributed Arithmetic based FIR Filter Architectures . . . . .	92
6.3.1 Systolic RAM DA-FIR Filter Architecture . . . . .	93
6.3.2 Shared RAM DA-FIR Filter Architecture . . . . .	94
6.3.3 Proposed ROM-MUX DA-FIR Filter Architecture . . . . .	95
6.3.4 Proposed Systolic ROM DA-FIR Filter Architecture . . . . .	96
6.4 Analysis of DA-FIR Filter Architectures . . . . .	97
6.4.1 Hardware Complexity . . . . .	97
6.4.2 Time complexity . . . . .	100
6.4.3 Performance analysis . . . . .	102
6.5 Hardware Implementation . . . . .	107
6.6 Conclusions . . . . .	111
<b>7 Conclusions and Future Scope</b>	<b>113</b>
7.1 Conclusions . . . . .	113
7.2 Future Scope . . . . .	116
<b>Appendices</b>	<b>117</b>
<b>A ASIC Synthesis Results</b>	<b>117</b>
A.1 Area Report . . . . .	118
A.2 Power Report . . . . .	119

---

A.3 Timing Report . . . . .	120
<b>Publications</b>	<b>122</b>
<b>Bibliography</b>	<b>124</b>

# List of Figures

2.1	Block diagram of a Radio Communication System . . . . .	9
2.2	Various Spectrum Allocation methods employed in Radio Communication Systems . . . . .	10
2.3	Communication of Mobile Stations with the Base Station in Traditional Radios .	11
2.4	Communication of Mobile Stations with the Base Station in Modern Radios . .	11
2.5	Traditional Hardware Radio Architecture . . . . .	12
2.6	Architecture of an Ideal Software Defined Radio . . . . .	13
2.7	Architecture of a practical Software Defined Radio . . . . .	14
2.8	Block Diagram of a Digital Front-End (DFE) of a SDR Transmitter . . . . .	16
2.9	Example of formation of a Wideband Digital IF Signal . . . . .	17
2.10	Architecture of a Uniform bandwidth Channelizer . . . . .	18
2.11	Architecture of a Non-Uniform bandwidth Channelizer . . . . .	19
3.1	Various Channelization approaches Proposed in the literature . . . . .	22
3.2	Channelizer based on Per-Channel approach . . . . .	23
3.3	Channelizer based on Discrete Fourier Transform Filter Bank approach . .	23
3.4	Channelizer based on Pipelined Frequency Transform approach . . . . .	25
3.5	Channelizer based on Frequency Response Masking Filter Bank approach . . .	25
3.6	Channelizer based on Coefficient Decimation Filter Bank approach . . . . .	27

---

3.7	Digital IF Stage Architecture . . . . .	30
3.8	Conventional SRC Architecture . . . . .	31
3.9	Variable Digital Filter based SRC Architecture . . . . .	32
3.10	Sample Rate Conversion structures . . . . .	33
3.11	Sample Rate Conversion Architectures proposed in the literature . . . . .	33
3.12	Direct Form FIR Structure for Decimation . . . . .	34
3.13	Application of Duality and Transposition in FIR filter based SRC Architecture .	35
3.14	Gain Compensation and Interpolation methods Proposed in the literature . . .	40
4.1	Block Diagram of Programmable Sample Rate Conversion Filter for SDR Receivers	48
4.2	Sample Rate Conversion by Large Integer Rates . . . . .	50
4.3	Structure of the CIC Filter to support Multi-standard Radio Communications .	52
4.4	Structure of Multi-stage CIC Filter . . . . .	53
4.5	Structure of the CIC filter with Gain normalized to unity . . . . .	53
4.6	Frequency Response of CIC Filter for GSM900 Radio Standard ( $R = 384$ ,simulated in MATLAB) . . . . .	55
4.7	Frequency Response of Gain Compensation Filter designed for GSM900 Radio Standard . . . . .	58
4.8	Structure of the Farrow Filter . . . . .	58
4.9	Frequency response of Lagrange Cubic Polynomial Interpolation Filter . . . .	60
4.10	Frequency Response of Proposed Modified Joint compensation and Interpolation Filter and Joint Compensation and Interpolation Filter . . . . .	63
4.11	Discrete Compensation and Interpolation Filter Architecture for Multi-standard Radio Receiver . . . . .	66
4.12	Proposed Joint Compensation and Interpolation Filter Architecture for Multi-standard Radio Receiver . . . . .	67

---

---

5.1	Architecture of SVD based Variable Digital Filter . . . . .	75
5.2	Index V/s Singular Value plot . . . . .	80
5.3	Architecture of SVD-VDF based Joint Compensation and Interpolation Filter	80
5.4	Frequency Response of various CIC Compensation and Interpolation Filters . .	82
6.1	Block Diagram of Distributed Arithmetic Based FIR Filter . . . . .	92
6.2	Various DA-FIR Filter Architectures proposed in the literature . . . . .	93
6.3	Block Diagram of Systolic-RAM DA-FIR Filter Architecture . . . . .	93
6.4	Block Diagram of Shared-RAM DA-FIR Filter Architecture . . . . .	95
6.5	Block Diagram of Proposed ROM-MUX DA-FIR Filter Architecture . . . .	96
6.6	Block Diagram of Proposed Systolic ROM DA-FIR Filter Architecture . .	97
6.7	Functional diagram of DA-FIR Filter Architectures . . . . .	98
6.8	Order of the Filter versus Hardware Complexity (in terms of area of Full adder)	104
6.9	Histogram plot for Architecture versus Area (in terms of area of Full adder) . .	106
6.10	ASIC Synthesis Results using TSMC CMOS 90nm library . . . . .	110

## List of Tables

3.1	Comparison of various Channelizers proposed in the literature . . . . .	28
3.2	Hardware and Computational Complexity of FIR Filter based SRC Structures proposed in the literature . . . . .	36
3.3	Hardware and Computational Complexity of CIC Compensation Filters proposed in the literature . . . . .	43
4.1	Required SRC Ratio to support Multi-standard Radio Communications . .	48
4.2	Computation of SRC Factors to support Multi-standard Radio Communications . . . . .	49
4.3	Performance Comparison of CIC Filter structures on Virtex-6 FPGA . . .	54
4.4	Computation of CIC Gain Droop for various Radio Standards . . . . .	55
4.5	CIC Gain Droop Compensation and Fractional Rate Interpolation Factors for four Radio Standards [88] . . . . .	56
4.6	Comparison of Peak Gain Frequency (in MHz) of Existing and Proposed Joint Compensation and Interpolation Filters . . . . .	64
4.7	Comparison of Peak Gain Error (in dB) of Existing and Proposed Joint Compensation and Interpolation Filters . . . . .	65
4.8	Hardware Complexity Comparison of CIC Compensation and Interpolation Filters implemented on Kintex-7 XC7K325t-2ffg900 FPGA . . . . .	68

---

5.1	Specifications for Gain Droop Compensation and Fractional Rate Interpolation Factors for various Radio Standards . . . . .	76
5.2	CIC Gain Compensation achieved with different methods(in dB) . . . . .	83
5.3	Hardware Complexity Comparison of DDC-SRC based on CIC filters (in terms of MAC units) . . . . .	84
5.4	Hardware Complexity Comparison of DDC-SRC based on CIC Filters implemented on Kintex-7 XC7K325t900-2FFG FPGA . . . . .	85
6.1	Hardware Complexity of DA-FIR Filter Architectures . . . . .	99
6.2	Computation of Critical Path and Latency of DA-FIR Filter Architectures	101
6.3	Comparison of Hardware Complexity of DA-FIR Filter with $N = 64(65\text{-Tap})$ , $L = 16\text{-bit}$ , $M = 4$ , $b = 1$ (Radix-2) . . . . .	102
6.4	Hardware Complexity of DA-FIR Filter Architectures for different Filter Orders, 16-bit Word-length and $M.b = 4$ in terms of Area of Full adder . .	103
6.5	Percentage Reduction in Hardware complexity of Proposed ROM-LUT based DA-FIR Filter Architectures . . . . .	105
6.6	Hardware Complexity and Latency Comparison of DA-FIR Filter with $N = 64(65\text{-Tap})$ and $L = 16\text{-bit}$ . . . . .	106
6.7	Performance Comparison of DA-FIR SVD-based SRC Filter implemented on Kintex-7 XC7K325t900-2ffg FPGA . . . . .	108
6.8	ASIC Synthesis Results of DA-FIR SVD-based SRC Filter with TSMC CMOS 90nm Library . . . . .	109
6.9	Performance Comparison of ASIC Synthesis Results of Proposed DA-FIR SVD-based SRC filter . . . . .	111

## List of Abbreviations

SDR	Software Defined Radio
RF	Radio Frequency
IF	Intermediate Frequency
BB	Baseband
SRC	Sample Rate Conversion
CIC	Cascaded-Integrator-Comb
FPGA	Field Programmable Gate Array
SVD	Singular Value Decomposition
VDF	Variable Digital Filter
ASIC	Application Specific Integrated Circuit
TSMC	Taiwan Semi-conductor Manufacturing Company
CMOS	Complementary Metal Oxide Semi-Conductor
RAM	Random-Access-Memory
ROM	Read-Only-Memory
LUT	Look-Up-Table
DA	Distributed Arithmetic
FIR	Finite-Impulse-Response
Tx	Transmitter
Rx	Receiver
LAN	Local Area Network
PAN	Personnel Area Network

UL	Up-link
DL	Down-link
WiMAX	World Interoperability for Microwave Access
CDMA	Code Division Multiple Access
GSM	Global System for Mobile
MS	Mobile Stataion
BS	Base Station
ADC	Analog-to-Digital Converter
DAC	Digital-to-Analog Converter
SFDR	Spurious-Free Dynamic Range
DFE	Digital Front-End
AFE	Analog Front-End
PC	Per Channel
DFTFB	Discrete Fourier Transform Filter Bank
PFTFB	Pipelined Frequency Transform Filter Bank
FRMFB	Frequency Response Masking Filter Bank
CDFB	Coefficient Decimation Filter Bank
CORDIC	CO-ordinate Rotation DIgital Computer
DDC	Digital Down Converter
DDS	Direct Digital Synthesis
DSP	Digital Signal Processor
ACI	Adjacent Channel Interference
SOPOT	Sum-of-Powers-of-Two
PFIR	Programmable FIR
CFIR	Compensation FIR
VFD	Variable Fractional Delay
HBF	Half-band Filters
PAT	Pipelined Adder Tree
SAT	Shift Adder Tree
PPG	Partial Product Generator
HDL	Hardware Description Language
VHDL	Very High Speed Integrated Circuit HDL

# Chapter 1

## Introduction

Over the past two decades, there has been an exponential growth in the wireless industry due to the emergence of new wireless technologies at a faster pace. New radio access technologies have come into existence to meet the objectives of a specific application. Synchronization in spectrum allocation and access to multiple radio communication standards need a single common platform. Traditional hardware radios are inflexible to adapt to the changes that occur with the change in radio communication standards as they employ fixed spectrum allocation techniques and most of the signal processing tasks are performed on the analog signal with fixed hardware. Software Defined Radio (SDR) is a radio communication system, which employs dynamic spectrum allocation technique and replaces analog signal processing with digital signal processing. SDR receivers have become feasible due to advancements in the data converter technologies and digital signal processing techniques.

### 1.1 Motivation

Due to advances in wireless technology, there is a tremendous demand for high data rate applications such as video games, interactive video, video monitors, streaming multimedia, mobile TV, 3D services, and video sharing. To meet these requirements, radio communication standards are changing at a faster pace. Radio standards can be implemented on either hardware or software platform. In case of hardware implementation, the hardware needs to be redesigned with the change in radio communication standards. Re-

cent advances in technology led to the development of high sampling rate analog-to-digital converters (ADCs), enabling the designers to implement most of the radio functionality in software. A radio receiver consists of radio frequency (RF), intermediate frequency (IF) and baseband (BB) processing stage. In software radio receivers, RF stage translates the high frequency RF signal into a wideband IF signal. The wideband IF signal, which incorporates multi-standard radio channels, is digitized at a Nyquist rate. Due to the phenomenon of bandpass sampling, the narrowband radio channels get oversampled. This results in increase in the computational complexity, hardware complexity and power dissipation in the BB processing stage of SDR receiver. Hence, there is a need to design a sample rate converter in the IF stage to decimate the sampling rate of the narrow band radio channel by a sample rate conversion factor equal to its oversampling ratio. It may also be noted that a sample rate converter designed for SDR receiver must be reconfigurable to support existing or forthcoming new radio communication standards. It is required to reduce the reconfiguration overhead while designing SRC filter for SDR receivers. Hence, it is necessary to design an area efficient reconfigurable architecture for sample rate conversion in SDR receivers.

## 1.2 Research Objectives

The objective of our research is to design and implement a reconfigurable SRC filter for SDR receivers for processing wideband IF signals with low computational complexity, and tuneable spectral characteristics while achieving minimum reconfiguration overhead and low hardware complexity.

- To design sample rate conversion (SRC) filter with low hardware and low computational complexity for achieving SRC by large reconfigurable SRC factors. This is addressed by designing low complexity Cascaded-Integrator-Comb (CIC) filters. However, CIC filters introduce gain droop in the pass band of interest and cannot be employed to attain SRC by fractional rate change. Hence, it is required to design a gain compensation and interpolation filter to restore the gain droop and attain the required symbol rate for supporting multi-standard ratio communications. The functionality and performance have to be verified by implementing on Xilinx FPGA

prototype board.

- To design reconfigurable SRC filter with less reconfiguration overhead for supporting multi-standard radio communications in SDR receivers. This can be achieved by designing variable digital filters employing singular value decomposition technique (SVD-VDF) which results in fixed coefficient filters with tuneable spectral characteristics and minimum reconfiguration overhead. It is required to verify the functionality and performance of this SVD-VDF by implementing on Xilinx FPGA prototype board.
- To propose an area efficient architecture for an SVD-VDF employing distributed arithmetic (DA) algorithm. It is required to verify the functionality and performance of DA-based SVD-VDF by implementing on FPGA and ASIC technologies.

### 1.3 Thesis Contributions

The contributions of the thesis are summarized as follows:

- **Sample Rate Conversion Filter architecture for SDR Receivers** A low hardware complexity reconfigurable SRC filter architecture for SDR receivers is proposed to achieve required sample rate change as per the specifications of a radio standard into consideration. The performance of this proposed architecture is evaluated through theoretical analysis and hardware implementations. The contributions of this work are briefly described as:
  - Proposed an area efficient multiplier-less Cascaded-Integrator-Comb filter architecture to achieve SRC by large integer factors. SRC factors are easily reconfigurable due to their coefficient-less architectures.
  - A CIC gain droop compensation filter based on the inverse frequency response of the CIC filter is designed using filter design methods in MATLAB and cascaded. A fractional rate interpolation filter based on Lagranges polynomial interpolation formula is implemented as a Farrow structure.
  - A joint compensation and interpolation filter based on the approximation of the frequency response with a second order frequency domain polynomial is

designed and implemented as a Farrow structure. The frequency response of the designed filter is evaluated using MATLAB.

- Proposed a modified joint compensation and interpolation filter to eliminate the need of second order frequency domain approximation. In this method, the desired frequency response of the filter is expressed as the inverse frequency of the CIC filter and realized as a Farrow structure. The frequency response of the proposed filter is simulated in MATLAB and compared with the frequency response of the existing joint compensation and interpolation filter.
- Implemented the proposed gain compensation and interpolation filters on KINTEX-7 FPGA to verify its functionality. It is observed that the filters designed based on above methods needs to be redesigned with the change in the specifications of radio communication standard. This leads to the requirement of offline computations involving high reconfiguration overhead.

- **SVD-based Reconfigurable SRC filter for multi-standard SDR Receivers**

A variable digital filter based on singular value decomposition algorithm for the design of reconfigurable low pass filter is proposed in the literature. We have applied SVD-VDF to design a reconfigurable joint compensation and interpolation filter to achieve spectral characteristics of multiple radio communication standards. The contribution of this work is briefly described as follows:

- Proposed a Reconfigurable Joint compensation and interpolation filter: Architecture of variable digital filter for gain compensation and interpolation based on singular value decomposition (SVD-VDF) is developed. The proposed filter consists of a cascade of fixed coefficient sub-filters and multi-dimensional polynomials defined in terms of spectral parameters such as pass band edge frequency, stop band edge frequency etc. Reconfigurable spectral characteristics are achieved by reconfiguring these spectral parameters.
- Evaluated the frequency response of the proposed filter for four radio communication standards, namely, WiMAX, WCDMA, CDMA2000 and GSM900 in MATLAB and compared with the frequency response of the modified joint compensation and interpolation filter.

- The proposed reconfigurable SVD-VDF for joint compensation and interpolation is implemented on KINTEX-7 FPGA to verify its functionality and evaluate the hardware complexity. It is observed that the proposed filter achieves flexible reconfiguration when compared with the existing joint compensation and interpolation filter.
- **Area Efficient Architecture for SVD based Reconfigurable SRC filter** Distributed arithmetic based FIR (DA-FIR) filter architectures employing RAM-LUTs are proposed in the literature to eliminate the need of multipliers in the design of FIR filters. In this work, we are proposing ROM-LUT based DA-FIR filter architectures for the design of sub-filters in SVD-VDF based reconfigurable SRC filters. The performance of these architectures is evaluated through analytical computations and hardware implementations. The contributions of this work are briefly described as:
  - **Systolic RAM-LUT based DA-FIR Filter Architecture:** A Systolic RAM-LUT based DA-FIR filter architecture is developed to implement the sub-filters in the design of SVD-VDF based reconfigurable SRC filters. Analytical computation of hardware complexity shows that the systolic RAM-LUT based DA-FIR filter architecture requires more hardware.
  - **Shared RAM-LUT based DA-FIR Filter Architecture:** A shared RAM-LUT based DA-FIR filter architecture employing RAM-LUTs and multiplexers is developed to reduce the high hardware complexity observed in the Systolic RAM-LUT based DA-FIR filter architecture. The approximate values of area complexity computed analytically shows that a RAM-LUT occupies more area when compared with the ROM-LUT. However, ROM-LUT based architectures are unsuitable to implement filters with variable coefficients. Since SVD-VDF filters employs fixed coefficient sub-filters in their design, the hardware complexity can be reduced by employing ROM-LUTs based DA architectures.
  - **Proposed ROM-MUX DA-FIR Filter Architecture:** The high hardware complexity of shared RAM-LUT based DA-FIR filter architecture is reduced by replacing RAM-LUTs with ROM-LUTs. The contents of the ROM-LUTs are shared by employing multiplexers. Analytical comparsion of hardware complexity of a multiplexer and a ROM-LUT shows that the multiplexer requires

more hardware.

- **Proposed Systolic ROM DA-FIR Architecture:** A systolic ROM-LUT based DA-FIR architecture is developed to eliminate the need of multiplexers and reduce the hardware complexity.
- Implemented SVD-VDF based joint compensation and interpolation filter employing various DA-FIR architectures on KINTEX-7 FPGA to verify its functionality.
- The hardware complexity of the various DA-FIR filter architectures is computed in terms of area of full adder and verified through hardware implementations. The netlist of proposed DA-FIR filter architectures is also synthesized using Synopsis Design Vision compiler with TSMC CMOS 90nm technology library for performance evaluation.

## 1.4 Thesis Organization

The rest of the thesis is structured as follows:

**Chapter 2** presents a brief overview on critical design aspects of Software Defined Radio.

**Chapter 3** presents the survey of channelization and sample rate conversion architectures available in the literature to design digital front-end in SDR receivers.

**Chapter 4** presents the design of SRC filter based on CIC filter with a cascade of modified gain droop compensation and interpolation filters. Analysis and hardware implementation results of these SRC filters for supporting multi-standard radio communication systems is presented.

**Chapter 5** presents the design of a reconfigurable SRC filter employing Singular Value Decomposition based Variable Digital Filter (SVD-VDF) technique to support multi-standard radio communications. Analysis and hardware implementations followed by the comparison of results with the existing works are performed.

**Chapter 6** presents the design of reconfigurable SRC filter employing distributed arithmetic to achieve area-efficient architecture is presented. The analysis and hardware implementations of this proposed area-efficient architecture are presented along with performance comparison with the available architectures.

---

**Chapter 7** draws conclusions from the earlier chapters and concludes the thesis.

## 1.5 Conclusions

In this chapter, we have presented a brief overview of the entire research work along with the motivation behind this research and its objectives. The next chapter presents an overview of the SDR concept and the critical stages in the design of SDR receivers are identified and presented.

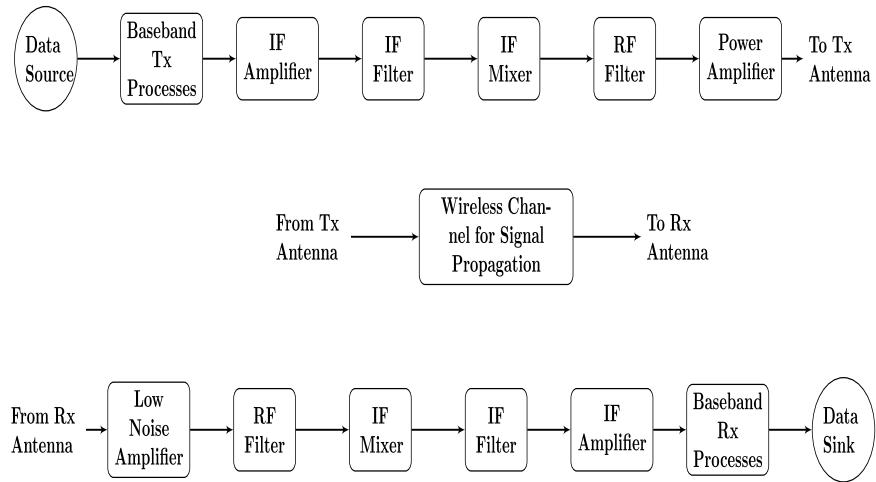
# Chapter 2

## Software Defined Radio

In this chapter the concept of radio communication systems and the implementation transition from traditional hardware radio to Software Defined Radio (SDR) is presented. The importance of Radio communication systems, different spectrum allocation techniques and their implementation platforms, disadvantages of traditional hardware radio communication systems are presented along with the necessity of SDRs. The concept of an ideal SDR along with the limitations in its implementation is presented. Solution to overcome these limitations is a practical SDR which comprises of two stages, namely, analog front-end stage and digital front-end stage. Since, the focus of our research is in the design of reconfigurable digital front-end, we emphasise on the design of digital front-end stage for SDR transceivers.

### 2.1 Radio Communication Systems

The efficiency and flexibility of the Radio Frequency wireless communication systems, and advancements in wireless technologies have led the researchers to explore their applications in different aspects of day-to-day life. Voice services and data services are the two important applications of RF communication systems. In the present state of art, there is a huge demand for high data rate applications such as video gaming, live multimedia streaming, mobile television, video sharing etc, when compared to the voice communications. These high data rate requirements led to the changes in radio communication standards at a faster pace.



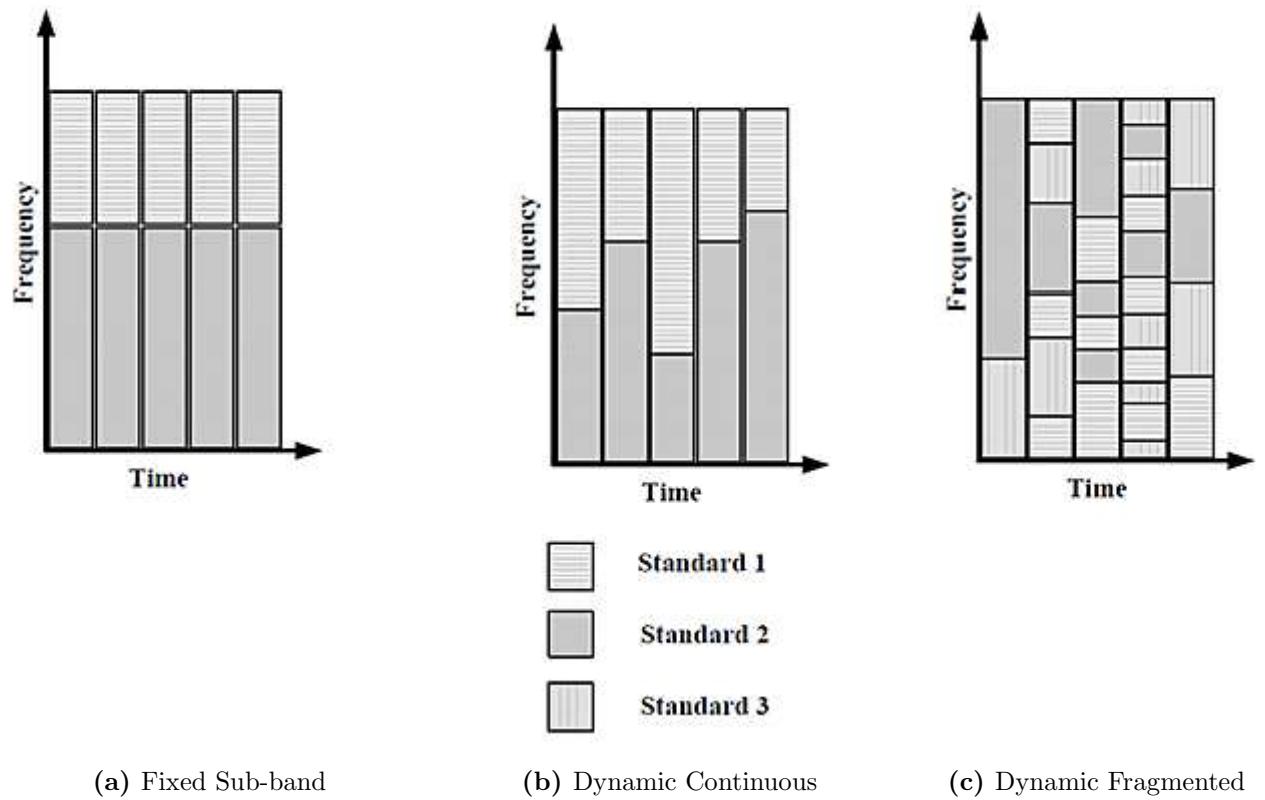
**Figure 2.1** Block diagram of a Radio Communication System

Fig. 2.1 shows the typical architecture of a wireless radio communication system which transfers information between radio transmitter (Tx) and radio receiver (Rx) on radio waves employing free space as a channel for signal propagation. The information source provides information such as an audio signal, a video signal, data or a combination of three. On the transmitter side, the information signal is amplified, modulated and transformed into a RF signal such that it is made suitable for radio transmission. On the receiver side, the modulated information signal is isolated from the received RF signal and the information is retrieved by the process of decoding and demodulation.

A mobile radio communication system can be implemented either on a traditional hardware platform or a software platform. The hardware design for mobile communications depends on the spectrum allocation methods employed by them. Fig.2.2 [1] shows the plot of frequency sub-band allocation with respect to time. Different techniques employed for the efficient spectrum utilization are:

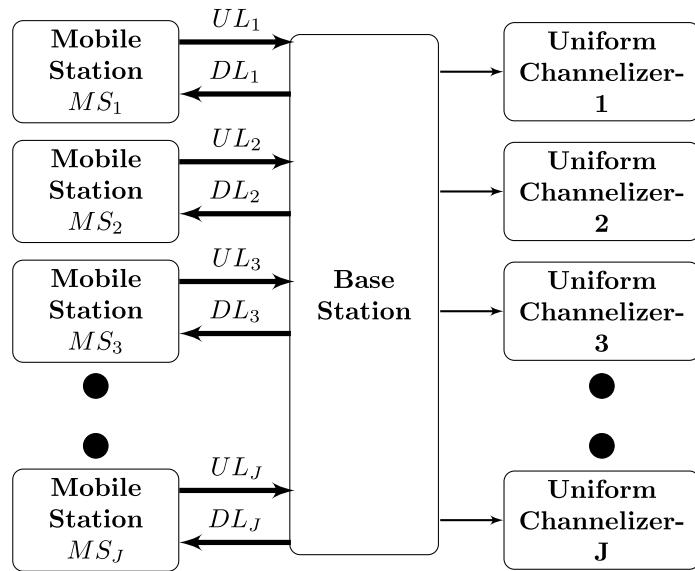
- Fixed Sub-band allocation
- Dynamic Continuous Sub-band allocation
- Dynamic Fragmented Sub-band allocation

It may be observed that the width of the sub-band in fixed sub-band allocation technique (See Fig.2.2a.) is fixed at any given interval of time, irrespective of the num-

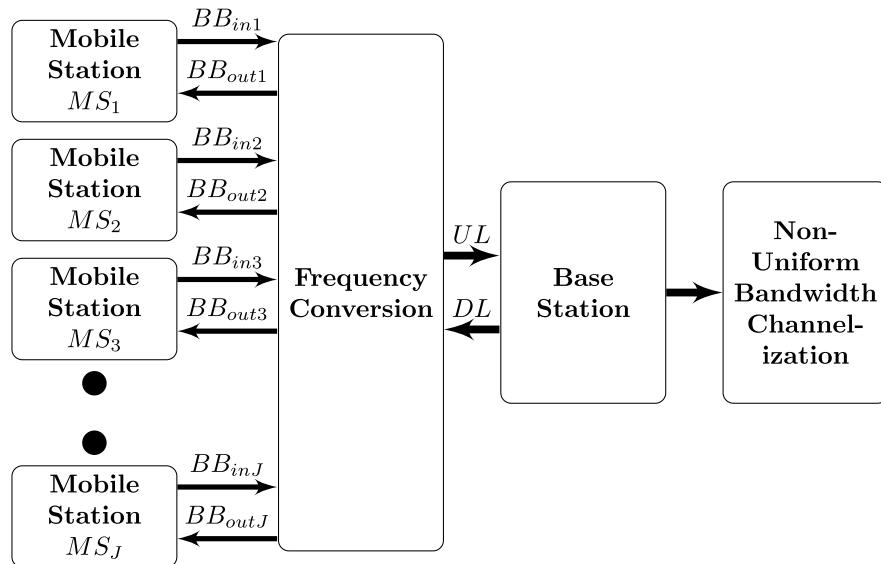


**Figure 2.2** Various Spectrum Allocation methods employed in Radio Communication Systems

ber of users accessing the spectrum. This leads to inefficiency in spectrum management. The spectrum efficiency is improved in dynamic contiguous sub-band allocation technique (See Fig.2.2b.). In this technique, at any given interval of time if the less number of users access the spectrum of a given radio standard then the remaining portion of the spectrum can be made available for the users of adjacent radio standard. In addition, the width of the sub-band is not fixed and the sub-band for multiple channels of a particular radio standard is contiguous. The spectrum efficiency can be further improved by employing dynamic fragmented sub-band allocation (See Fig.2.2c.). In this technique, no portion of the spectrum is fixed for the user of a particular radio standard. Any portion of the spectrum can be utilized by the user of any radio standard. It may be noted that fixed spectrum allocation technique can be implemented with traditional hardware radio while the dynamic fragmented sub-band allocation technique necessarily requires the implementation of a software defined radio.



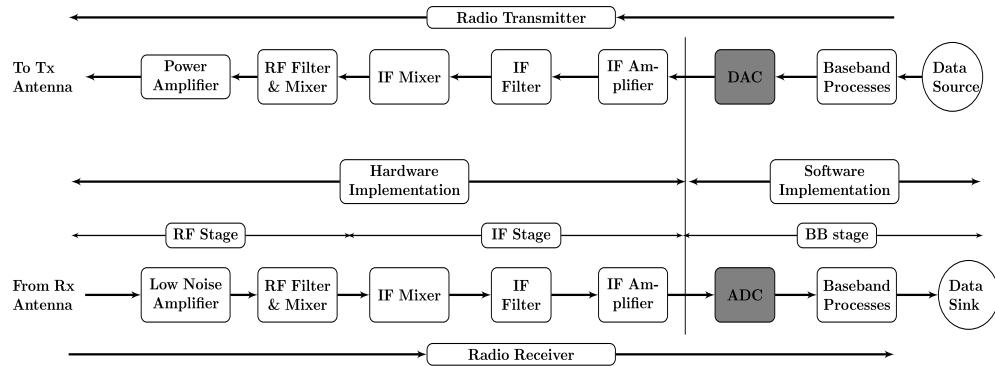
**Figure 2.3** Communication of Mobile Stations with the Base Station in Traditional Radios



**Figure 2.4** Communication of Mobile Stations with the Base Station in Modern Radios

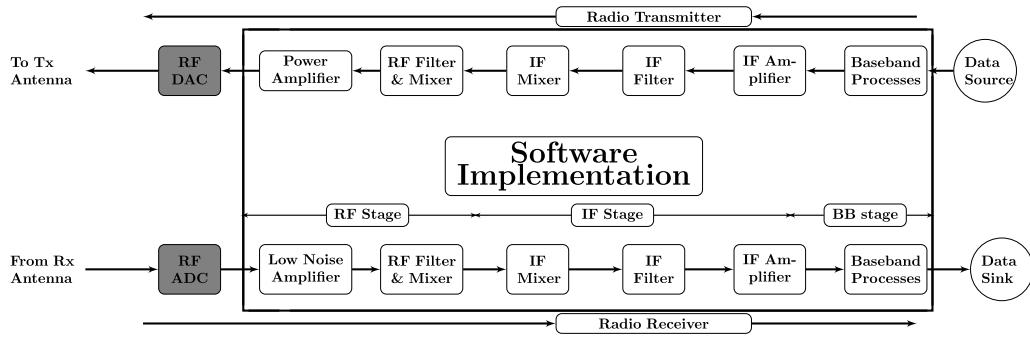
Fig.2.3 shows the communication mechanism employed for the implementation of a traditional hardware radio. In this system, mobile stations operating on different radio communication standards communicate with the base station on their dedicated uplink/downlink (UL/DL) channel. A dedicated RF channelizer is required to support each radio channel. The number of RF channelizers increases with increase in the number of radio communication standards resulting in hardware complexity. Fig. 2.4 shows a

solution to reduce the hardware complexity. In this system, the baseband (BB) signals of mobile stations operating on various radio communication standards communicate with the base station over a single uplink/downlink channel and employ a single non-uniform channelizer which reduces the hardware resources.



**Figure 2.5** Traditional Hardware Radio Architecture

Fig.2.5 shows the architectural implementation of a traditional hardware radio transceiver employing a single uplink/downlink channel. It is divided into three stages, namely, Radio Frequency (RF) processing stage, Intermediate Frequency (IF) processing stage and Baseband (BB) processing stage. On the transmitter side, the BB stage performs tasks such as coding and modulation of information signal, IF stage performs filtering and frequency up-conversion of the modulated signal, and RF stage performs the conversion of the IF signal into high frequency RF signal for radio transmission. On the receiver side, the RF stage converts the incoming RF signal into IF signal, IF stage performs the tasks of IF filtering and frequency down conversion, and BB stage performs decoding and demodulation. In traditional hardware radios, since the operating frequency of available data converters was low, the high frequency processing such as RF and IF signal processing is carried out on analog signal using fixed hardware while the low frequency baseband signal processing is implemented in software. Due to this, the hardware designed for one radio standard becomes obsolete and requires to redesign for incorporating a new radio standard in the system. Software defined radio is a solution to address this problem.



**Figure 2.6** Architecture of an Ideal Software Defined Radio

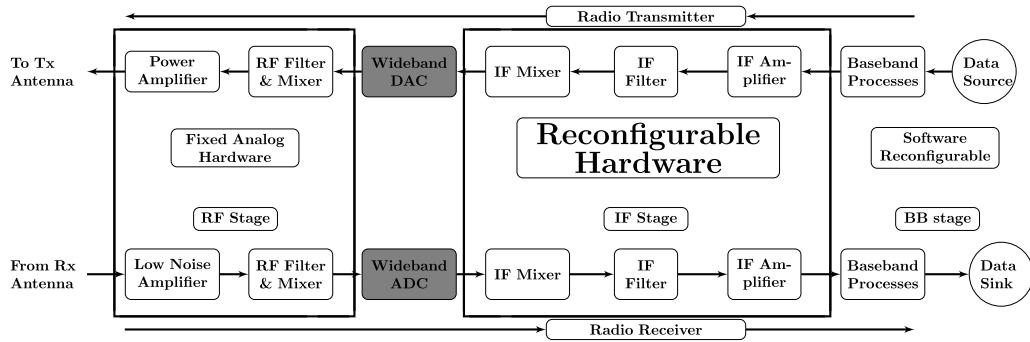
## 2.2 Ideal SDR

The concept of SDR is first introduced by Joseph Mitola in 1991 [2]. The driving forces behind the development of SDR are summarized as follows:

- A mobile phone should provide World Roaming.
- The performance features of the radio telephone must be combined with the functionality of personnel area network (PAN) and local area network (LAN).
- Lower the production cost by implementing multiple standards on a single radio platform. An improvement to the present radio can be incorporated via software upgrades and the radio can be made future proofed.

An SDR is a radio communication device in which some or all the physical layer functions are software defined. Reconfigurability, adaptability and multi-mode operations such as multiple air interfaces with variable quality of service are the advantages of SDR when compared to the hardware radios. Recent advancements in the VLSI technology led to the development of high speed data converters operating at high frequencies. Due to this, data converters are placed as close as possible towards the antenna and all the processing tasks are carried out in software.

Fig.2.6 shows the architecture of an ideal SDR with the data converters placed immediately after the antenna and a radio transceiver, i.e., RF processing to BB processing can be implemented in software. The parameters to be considered in the design of a



**Figure 2.7** Architecture of a practical Software Defined Radio

SDR transmitter are output power level, power control range and spurious emissions. Similarly, input sensitivity, maximum expected input signal and blocker specifications are the parameters to be considered in the design of a SDR receiver.

Data converters are the limiting factors in the implementation of an ideal SDR. On the transmitter side, SDRs require digital-to-analog converters (DACs) operating at radio frequencies resulting in high power dissipation and increase in spurious emissions [3]. Similarly, an ideal SDR receiver requires digitization of RF signals with high sampling rate analog-to-digital converters (ADCs) at the antenna resulting in high power dissipation in the subsequent stages of SDR receiver [4]. In addition, the resolution of the converter decreases by one bit for every doubling of the sampling frequency. This results in tight blocker specifications for the design of SDR receiver due to decrease in the spurious free dynamic range (SFDR) of the converter [5]. A practical SDR architecture provides a solution to overcome the limitations of an ideal SDR.

Fig.2.7 shows the architecture of a practical SDR obtained by placing the data converters at IF stage instead of RF stage. Since, IF signal processing is carried out relatively at a lower frequency when compared to the RF frequency, the specifications of the data converters are relaxed. The architecture of a practical SDR can be divided into two stages namely, analog front-end stage to perform high frequency RF signal processing and the digital front-end stage to perform relatively low frequency IF and baseband signal processing.

On the transmitter side, the narrow band information signals are digitally coded and modulated in the baseband processing stage. These narrow band digitally modulated

signals are up converted and channelized to form a wideband digital IF signal and converted into analog signal using wideband DACs in the IF stage. The frequency of the wideband IF signal is further up converted into high frequency RF signal in the RF stage and radiated through antenna.

On the receiver side, the frequency of the high frequency RF signal received from the antenna is down converted into a relatively low frequency wideband IF signal in the RF stage. The wideband IF signal is digitized using wideband ADCs and multiple narrow band radio channels are extracted from the wideband digital IF signal using digital down converters in the IF stage. Demodulation and decoding is carried out in the baseband processing stage of SDR receiver.

In the architecture of a practical SDR, the high frequency RF signal processing is implemented on fixed hardware employing analog signal processing techniques and the very low frequency baseband signal processing stage is implemented in software. The IF stage of SDR acts as a interface between RF processing stage and baseband processing stage and its frequency of operation lies between the frequency of operation of RF and baseband. The frequency of IF signal is selected such that it can incorporate multiple radio channels with variable bandwidths ranging from very wide to very narrow. Digital IF stage is implemented by employing special purpose digital signal processor or reconfigurable hardware (field programmable gate arrays (FPGAs)), as it involves high power dissipation at high sampling rates and cannot be implemented in software. Since, RF processing is implemented on fixed hardware and baseband processing can be easily reconfigured through software, it is required to design a reconfigurable hardware for digital IF stage. The next section presents the tasks carried out in the digital front-end stage of a SDR transceiver to support multi-standard radio communications.

## 2.3 Digital Front-End

In the architecture of a practical SDR, digital front-end (DFE) stage is placed apart from the antenna. This section presents the functionality of DFE stage in the design of practical SDR transmitters and receivers based on the characteristics of signals present at their input and output.

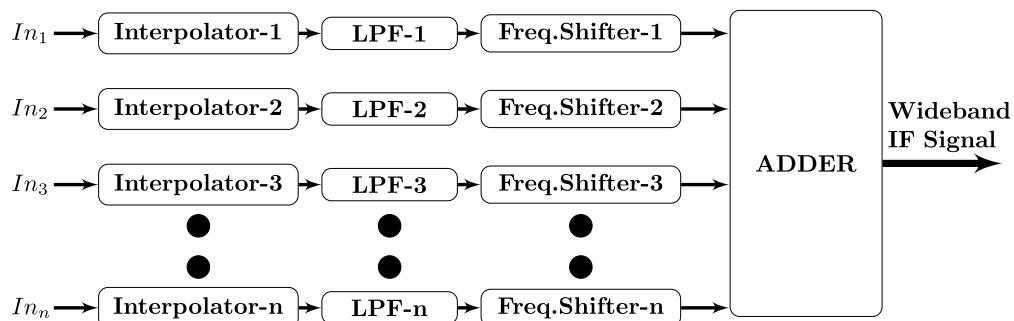
---

### 2.3.1 SDR Transmitter

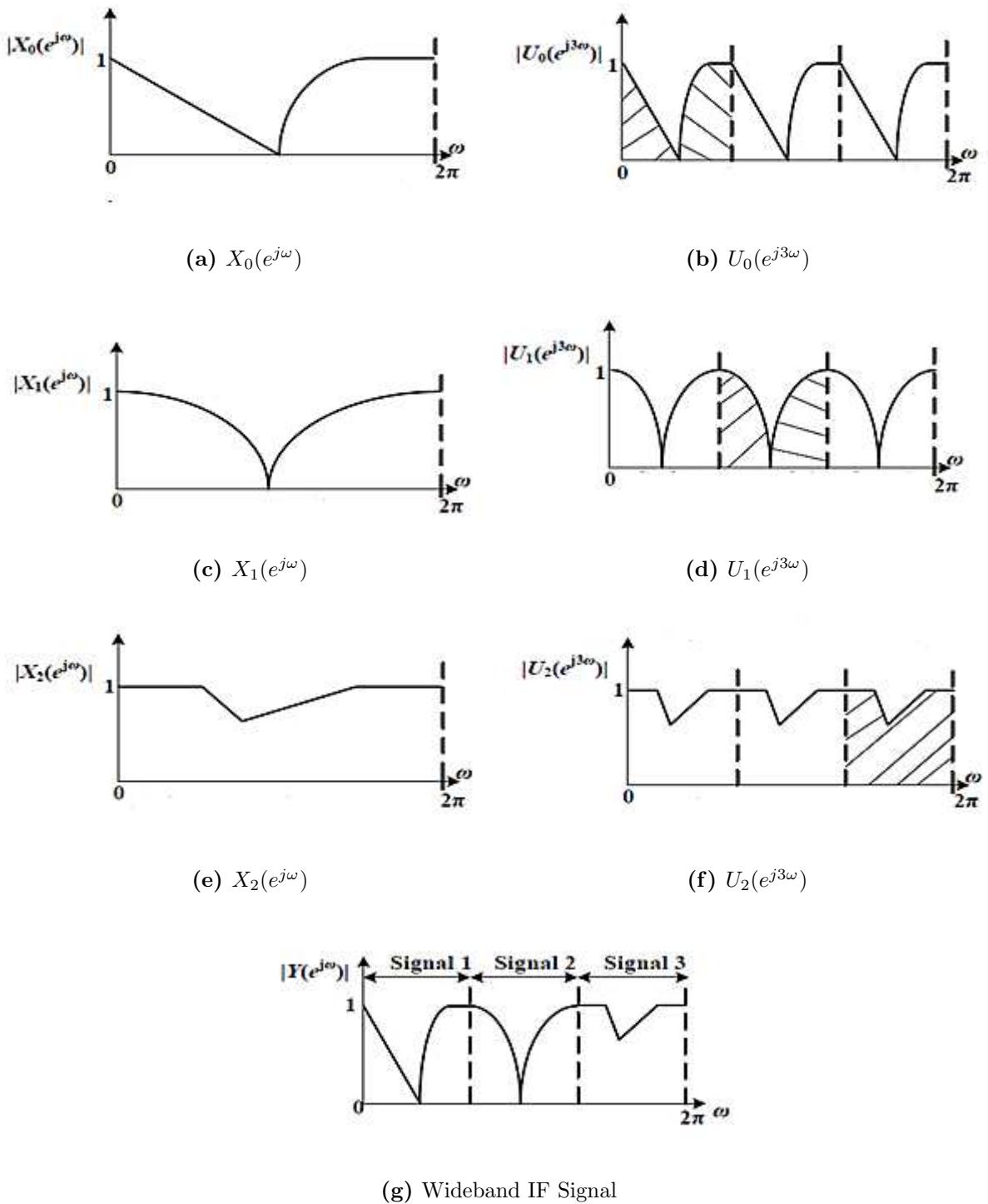
The function of DFE stage at the transmitting end is to form a wideband IF signal from the narrow band radio channels making them suitable for radio transmission. Fig.2.8 shows the block diagram of DFE of a SDR transmitter. It consists of interpolators, anti-imaging low pass filters (LPFs), frequency shifters and adder to generate a wideband IF signal. This wideband digital IF signal is converted into analog signal, frequency up converted to radio frequency and transmitted through antennas.

The bandwidth of the IF signal is selected based on the bandwidth supported by the data converters. The number of channels supported in a fixed bandwidth IF signal is inversely proportional to the channel bandwidth of the baseband signals. The method to generate a wideband IF signal from multi-standard baseband signals is explained as:

- Interpolate the sampling rate of the band-limited channels of multi-standard air interfaces by factor  $I$  such that it is equal to the Nyquist sampling rate of the wideband IF signal. This produces multiple copies of spectral images in the spectral domain.
- Except at DC frequency, reject  $(I - 1)$  spectral images of the interpolated multi-standard radio signals by employing anti-imaging low pass filters.
- Shift the centre frequency of the interpolated baseband signals from DC frequency to non-zero centre frequency such that the spectral image of one radio standard do not overlap with the spectral image of another radio standard.



**Figure 2.8** Block Diagram of a Digital Front-End (DFE) of a SDR Transmitter



**Figure 2.9** Example of formation of a Wideband Digital IF Signal

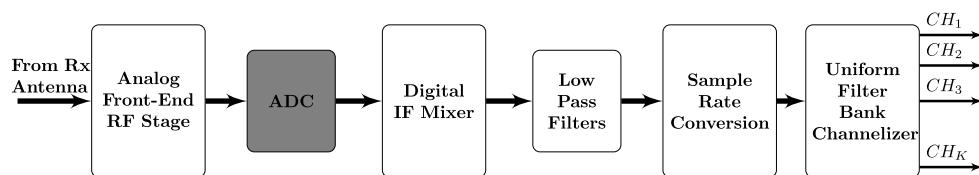
- Add these interpolated and frequency shifted signals to form a wideband IF signal.

Fig.2.9 [1] shows an example of generation of a wideband IF signal formed from three narrow band signals. Fig. 2.9a, 2.9c and 2.9e shows the band-limited spectrum of

narrow band signals  $x_0(n), x_1(n)$  and  $x_2(n)$ , respectively. Interpolation of these narrow band signals by a factor of three produces three images in the spectral domain as shown in Fig.2.9b, 2.9d and 2.9f. The two spectral images with non-zero center frequency are rejected and the center frequency of the remaining spectral component is shifted from zero frequency to non-zero center frequency and added in a manner such that a composite wideband IF signal is formed (See Fig.2.9g). This signal is further upconverted into a RF signal and transmitted through antennas.

### 2.3.2 SDR Receiver

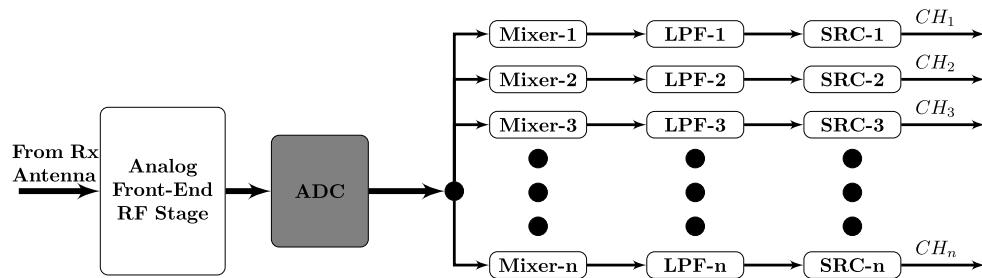
The RF signal received at the antenna is down converted into a wideband IF signal. The function of DFE stage at the receiving end is to digitize this wideband IF signal at Nyquist rate by employing wideband ADCs and extract the narrow band radio channels from the wideband digital IF signal with required signal characteristics and sample rate. The digitized wideband signal comprises of several radio channels with channel-of-interest centered at an arbitrary carrier frequency. Channelizer shifts the required narrowband radio channel to baseband and removes interference due to adjacent channels by means of digital filtering. It is also required to carry out baseband processing at a required target rate of the air interface. Hence, sample rate conversion is necessary between analog/digital interface to the target rate of the air interface. Since, channelization and SRC both require filtering, they can be implemented together and considerable savings in hardware can be achieved.



**Figure 2.10** Architecture of a Uniform bandwidth Channelizer

Based on the channel bandwidths of radio channels, an SDR channelizer can be implemented as:

- Uniform bandwidth channelizer



**Figure 2.11** Architecture of a Non-Uniform bandwidth Channelizer

- Non-Uniform bandwidth channelizer

A uniform bandwidth channelizer is one which extracts several narrow band radio channels of uniform bandwidths or a single radio standard from the wideband IF signal. A non-uniform bandwidth channelizer is one which extracts narrow band radio channels of multiple radio standards with non-uniform bandwidths from a wideband IF signal. Fig.2.10 and 2.11 shows the architecture of a uniform bandwidth channelizer and a non-uniform bandwidth channelizer, respectively. The hardware complexity of the DFE stage is estimated by assuming the hardware complexity of the RF and baseband processing stage remains unaltered for both channelizers and compared. The hardware complexity of DFE stage with uniform bandwidth channelization is estimated as the sum of hardware complexity of mixer, low pass filter, SRC and a uniform filter bank channelizer. In non-uniform bandwidth channelizer the hardware complexity is reduced by eliminating the filter bank channelizer. However, it requires multiple copies of mixers, low pass filters and SRCs to extract channels of multiple radio standards with non-uniform channel bandwidths. The hardware complexity of a non-uniform bandwidth channelizer increases with increase in the number of channels, but it can be easily reconfigured to support multi-standard radio communications in contrast to the uniform bandwidth channelizer.

Different features required for the design of channelizer in SDR receivers are:

- As SDR is capable of processing any radio communication standard with different channel bandwidths, SDR receiver must be capable to extract non-uniform bandwidth channels.
- It must be capable of extracting a very narrow bandwidth channel from a wideband

IF signal.

- Reconfigurability of the same filter bank for new communication standard with minimum reconfiguration overhead i.e. requirement of ultimate reconfigurability.
- Low area, low computational complexity, low power and high speed.

Hence, it may be noted that the design of DFE stage of SDR receiver with the above features is more complicated compared to the SDR transmitter.

## 2.4 Conclusions

This chapter presents the SDR implementation of wireless communication systems. It is observed that the design of a DFE for SDR receiver is more complicated compared to the design of other stages. The next chapter presents the survey of various methods proposed in the literature to implement DFE stage of SDR receivers.

# Chapter 3

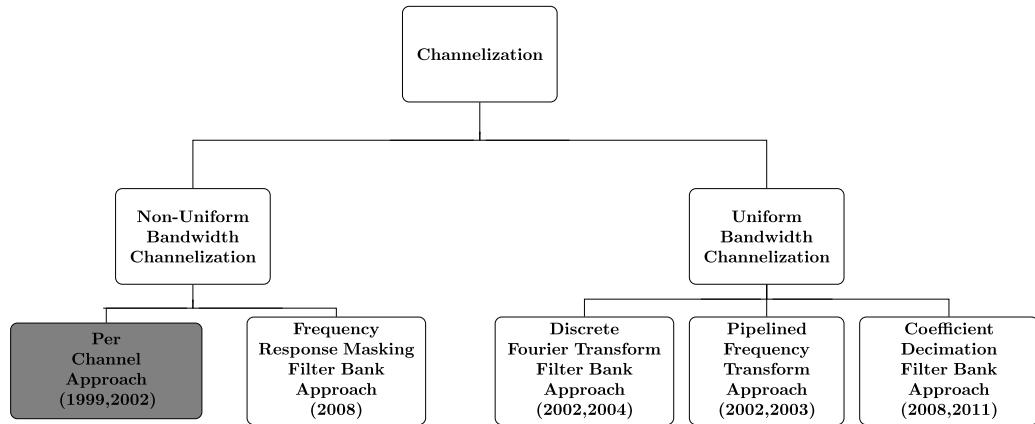
## Digital Front-End Stage of SDR Receivers

In this chapter, the various techniques proposed in the literature to carry out the two computational intensive tasks, channelization and sample rate conversion (SRC) in the digital front-end stage of SDR receivers are presented. The filtering operations involved in the channelization and SRC can be merged into a single filter to achieve reduction in the hardware. Design of SRC filters for SDR receivers requires extraction of non-uniform multi-standard radio channels. It may be noted that the filter specifications such as pass band edge frequency, stop band edge frequency change with the change in specifications of radio standard which leads to the requirement of variable digital filter (VDF) in SDR receivers. Various methods available in the literature to design VDF are presented. Finally, we present a brief review on multiplier-less architectures proposed in the literature for the implementation of digital filters to achieve reduction in hardware complexity, computational complexity, power dissipation and latency.

### 3.1 Channelization

An SDR receiver must be capable of extracting non-uniform bandwidth channels of multiple radio standards from a wideband digital IF signal with low hardware complexity and must be reconfigurable to support any existing or forthcoming radio standard with minimum reconfiguration overhead.

Fig.3.1 shows the various approaches proposed in the literature to design channelizers for SDR receivers. The design methodology of these approaches and their performance



**Figure 3.1** Various Channelization approaches Proposed in the literature

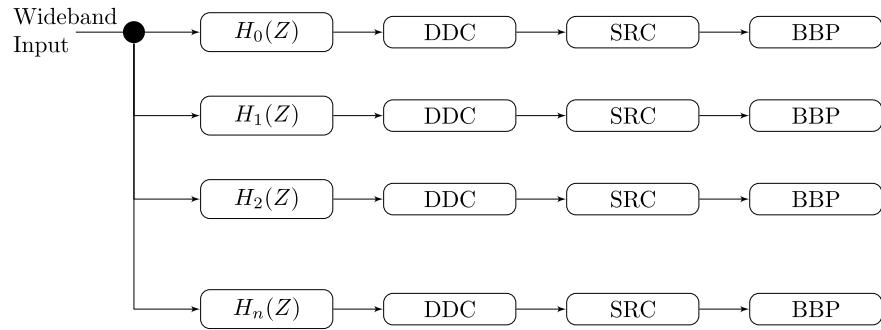
comparision is presented briefly in the following sub-sections.

### 3.1.1 Per-Channel Approach

Per-Channel (PC) approach is a straight forward technique for extraction of non-uniform bandwidth channels in SDR receivers [6]. This requires extraction of channel of interest through parallel arrangement of filters followed by digital down converter (DDC) and sample rate converters (SRC) per channel as shown in Fig.3.2. In an  $N$ -channel channelizer,  $H_0(z)$  is a low pass filter and the rest of the filters are band pass filters, if DDC is placed after filtering operation. In this architecture, filters operate at high sampling rate demanding high speed computations resulting in high power dissipation. This problem can be addressed by placing DDC-SRC prior to the channel filters resulting in the replacement of the band pass filters with low pass filters. It may be observed that the hardware complexity of this approach increases with number of channels being extracted.

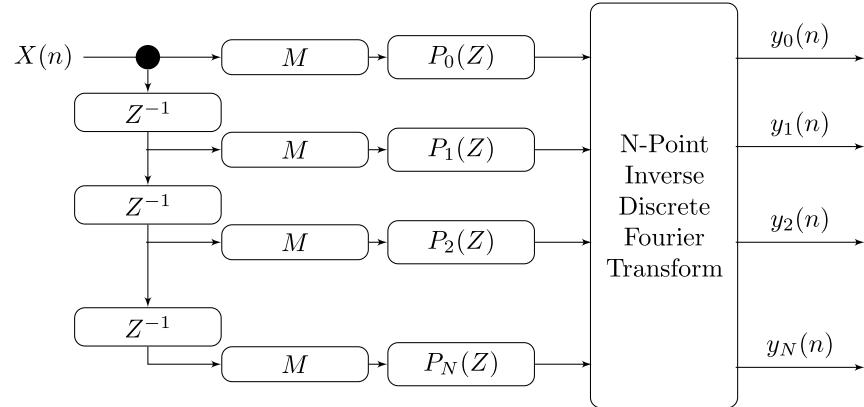
### 3.1.2 Discrete Fourier transform Filter Bank Approach

Filter banks based on discrete Fourier transform filter bank (DFTFB) is proposed as an alternate when the number of channels increases to a large number [7]. DFTFB is suitable for extracting channels of uniform bandwidth. Fig.3.3 shows the architecture



**Figure 3.2** Channelizer based on Per-Channel approach

of DFTFB based channelizer. The basic idea in this approach to extract  $N$ -uniform channels is to employ  $N$ -polyphase filters as modulating filter bank filters followed by  $N$ -point DFT/IDFT and down conversion by a factor  $N$ . However, DFTFB channelizers cannot be used for multi-standard radio receivers as they require more reconfiguration overhead due to the following reasons,



**Figure 3.3** Channelizer based on Discrete Fourier Transform Filter Bank approach

- The number of polyphase branches increases with the number of channels to be extracted. Apart from this, the design of filter coefficients in each polyphase branch with variable channels becomes tedious.
- Reconfiguration of DFT architecture if the radio communications changes from one standard to another standard.

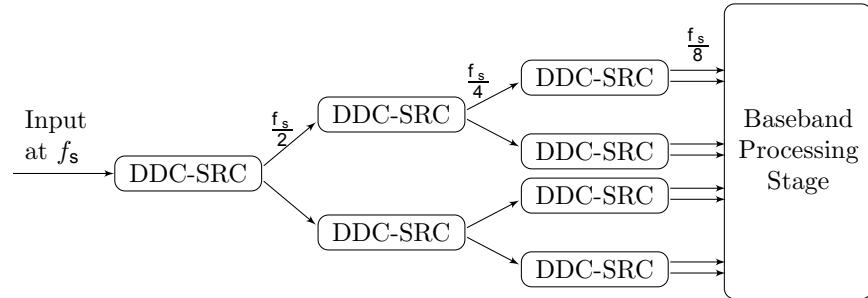
- A narrow band channel when extracted from a wideband IF signal requires a filter with high selectivity, thus increase in the order of channel filter and thereby increased computational complexity.

A filter bank channelizer employing a fixed DFTFB in the front end followed by a fractional sample rate converter and a low speed DFTFB in the backend can be used to extract channels of multiple radio standards. In this architecture, hardware optimization can be achieved only for fixed front-end DFTFB as the hardware required for backend DFTFB changes with the change in radio communication standard [8]. A channelizer based on modulated perfect reconstruction bank (MPRB) is proposed in [9]. In this approach, the filter bank consists of two sections, an analysis section and synthesis section. Analysis section generates sub-band signals which can be added up in the synthesis section to generate a wideband signal. Thus, the bandwidth of wideband signal is always an integer multiple of bandwidth of sub-band signal. This approach is not suitable for multi-standard software radio receivers, as the bandwidth of wideband signal may not be always an integer multiple of sub-band signal. In addition, the hardware complexity of filter banks [8], [9] is twice that of DFTFB channelizer.

### 3.1.3 Pipelined Frequency Transform Approach

A pipelined frequency transform (PFT) technique [10] is proposed to overcome the high reconfiguration overhead involved in the DFTFB channelizer. In this technique, a binary tree of DDC and SRCs are employed to split the signal into high and low frequency subbands, followed by splitting each half-band again till the desired channel of interest is extracted. Fig.3.4 shows the architecture employed in this approach. The output sampling rate at each level of tree is decimated by two, resulting in reduction of computation complexity the PC approach and DFTFB approach. However, the filter bank is slower in operation and suitable if and only if the wideband IF signal is power of two multiple of desired channel of interest. A tree quadrature mirror filter bank (TQMFB) technique [11] similar to PFT technique is employed to extract the signals in quadrature. A numerically controlled oscillator and interleavers placed at intermediate levels of tree structure in the PFT approach leads to fine tuneable PFT (TPFT) channelizer [12]. However the computational complexity of the TPFT is more than the PFT approach. Channelizers

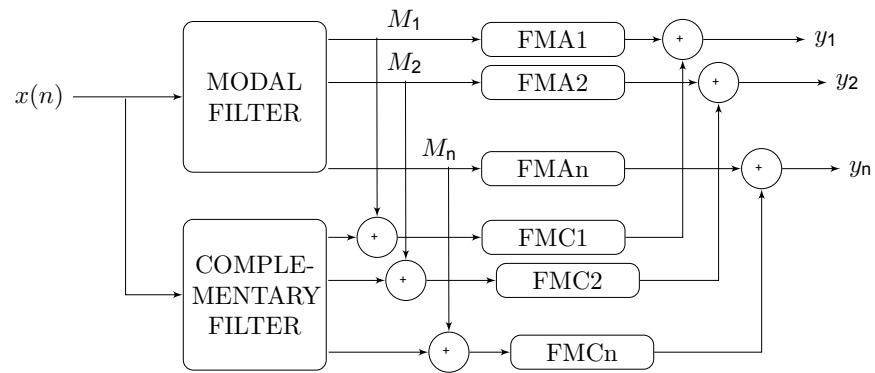
designed based on PFT techniques are unsuitable for wideband channelization required for SDR receivers due to the following reasons,



**Figure 3.4** Channelizer based on Pipelined Frequency Transform approach

- Wideband signal must be a power of two multiple of sub-band channel being extracted.
- A narrow bandwidth channel requires long chain of tree of DDC and SRCs which in turn contribute to more errors due to finite word-length effects at the final stage of decomposition.

### 3.1.4 Frequency Response Masking Filter Bank approach



**Figure 3.5** Channelizer based on Frequency Response Masking Filter Bank approach

Frequency response masking (FRM) approach was originally designed to meet sharp transition characteristics in filters with low computational complexity. For software radio receivers, filter banks with minimum overhead are required for supporting multi-standard

radio communications. This problem is addressed by proposing frequency response masking approach [13], [14], [15]. Direct realization of FIR filters for sharp transition band results in higher order filters resulting in increased computational complexity. In FRM approach, the sharp transition filter is realized as a combination of low complexity wide transition filters. The design of reconfigurable filter banks for multi-standard radio communications based on FRM technique is as follows.

The filter consists of four filters namely, modal filter and its complementary filter, two frequency response masking filters as shown in Fig.3.5. Let  $f_{p1}, f_{p2} \dots f_{pn}$  and  $f_{s1}, f_{s2} \dots f_{sn}$  be the pass band and stop band edges of the channel filters corresponding to the multi-standard radio communications. Let  $f_{ap}, f_{as}$  be the pass band and stop band edge of the low complexity modal filter (See Fig.3.5). In FRMFB approach, the specifications of the modal filter remain unchanged where as the required frequency specifications shall be achieved by designing the masking filters according to the specification of the radio standard.

The upsampling factors of the required channel filters,  $M_1, M_2 \dots M_n$  are obtained by solving the below equations,

$$\begin{aligned} f_{p1}M_1 - \lceil f_{p1}M_1 \rceil &= f_{p2}M_2 - \lceil f_{p2}M_2 \rceil \dots \dots \dots \quad (3.1) \\ &= f_{pn}M_n - \lceil f_{pn}M_n \rceil \end{aligned}$$

$$\begin{aligned} f_{s1}M_1 - \lceil f_{s1}M_1 \rceil &= f_{s2}M_2 - \lceil f_{s2}M_2 \rceil \dots \dots \dots \quad (3.2) \\ &= f_{sn}M_n - \lceil f_{sn}M_n \rceil \end{aligned}$$

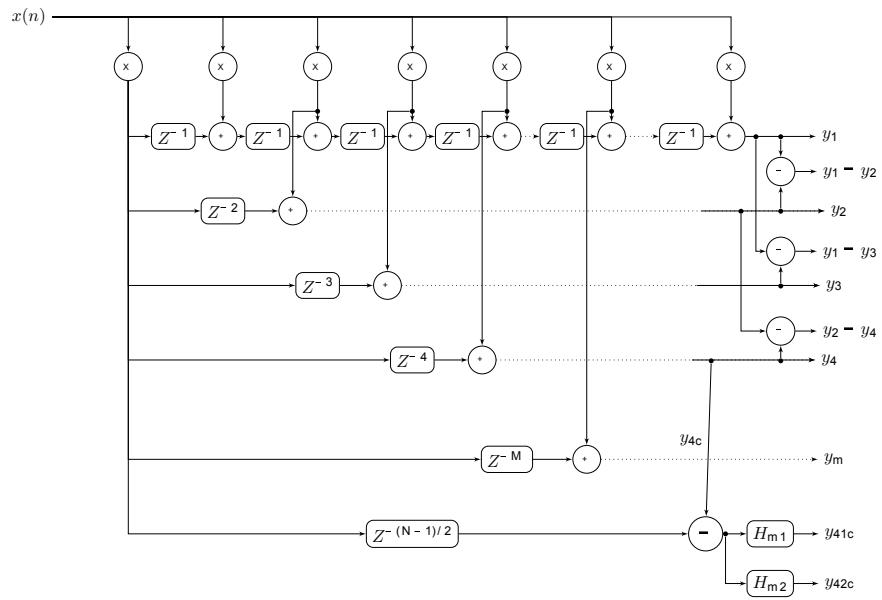
The architecture shares the same modal filter for various communication standards resulting in area and power savings with different values of  $M$ . The complementary of the modal filter is reconfigured just by reconfiguring the number of delay elements.

In FRMFB approach, the modal filter and its complementary filter remains unchanged. However, each mode of operation requires two masking filters of low complexity as the modal filters are wide transition band filters. In this approach, the frequency band of the modal filter and its complementary filter can be reconfigured at architectural level by changing the number of delay elements and at filter level by providing appropriate specifications for the masking filters. Any channel can be extracted by reconfiguring the

filters at either of these levels. FRMFB approach is suitable for extraction of non-uniform bandwidth channels as required in SDR receivers. This results in achieving reduction in hardware complexity due to low complexity involved in the design of wide transition band filters. The limitation of this approach is the requirement to redesign two masking filters with a change in radio standard specification leading to offline computations, a tedious task.

### 3.1.5 Coefficient Decimation Filter Bank Approach

Coefficient decimation filter bank (CDFB) approach [16]- [17] is proposed to eliminate the offline computations as required in FRMFB approach. Design of filter banks employing this technique achieves full control over the pass band widths and pass band locations when compared with the other filter banks proposed in the literature. The procedure for design of channelizers based on CDFB approach is as follows.



**Figure 3.6** Channelizer based on Coefficient Decimation Filter Bank approach

The design approach introduces  $(M - 1)$  zeros in the original filter with its impulse response coefficients,  $h(n)$ . This is same as discarding  $(M - 1)$  samples in the input sequence  $x(n)$  if it is down sampled by a factor of  $M$ . Fig.3.6 shows the architecture of CDFB and it may be noted that the filter architecture does not require design of any extra filter to incorporate a new radio channel of interest. Change of delay values leads

**Table 3.1** Comparison of various Channelizers proposed in the literature

Component	Per Channel [6]	DFTFB [7]	MPRB [8]	FRMFB [15]	CDFB [17]
Filter	$N_I L$	$L$	$2L$	$lm$	$L + l$
DDC	$N_I - 1$	—	—	$N_I - 1$	$N_I - 1$
Mod. Filters	—	$N_I^2$	$2N_I^2$	—	—
Sum	$N_I(L + 1) - 1$	$L + N_I^2$	$2L + 2N_I^2$	$lm + N_I - 1$	$L + l + N_I - 1$
R.Overhead	Low	High	High	High	High

to design of new filter. It may also be observed that a new channel of interest can be extracted by performing addition/subtraction operation on two channels provided the factor  $M$  for one channel is integral multiple for another channel. Hence, this approach is suitable for extraction of uniform bandwidth channels and it involves low reconfiguration overhead when compared to the DFTFB approach. However, tighter specifications can be met by designing appropriate frequency response masking filter as in the case of FRMFB approach resulting in high hardware and computational complexities.

### 3.1.6 Comparison of various channelization approaches

In this sub-section, the performance comparison of the various channelization approaches and their suitability for employing in SDR receivers is presented. Table 3.1 gives the computational complexity of channelizers [17]. The computation complexities of the channelizers are measured in terms of number of multiplications involved in FB channelization. Different operations involving multiplication are digital down conversion, channel filter and modulation of filters. Let  $N_I$  be the number of channels (of one communication standard) to be extracted from wideband IF signal,  $L$  be the number of non-zero coefficients in the design of channel filter,  $l$  be the number of non-zero filter coefficients in the design of modal filter in FRMFB and  $lm$  be the total number of non-zero coefficients for the design of modal filter and masking filters in FRMFB design,  $F_s$  is the sampling frequency.

It can be inferred from Table 3.1 that the computation complexity of PC approach is directly proportional to the number of channels,  $N_I$ , hence it is hardware inefficient.

The complexity of MPRB is exactly twice that of DFTFB because it consists of two DFTFB, one for analysis and the other for synthesis. However, DFTFB and MPRB FB are suitable if the number of channels to be extracted is power of two. FRMFB has least computation complexity, as the modal filters and masking filters are wide transition band filters. FRMFB has to be redesigned for each band pass channel which is a tedious task. CDFB can be employed for extraction of channels with uniform bandwidth with slightly increased computation complexity when compared with DFTFB and MPRB. However, as the number of channels increases CDFB is more efficient than DFTFB.

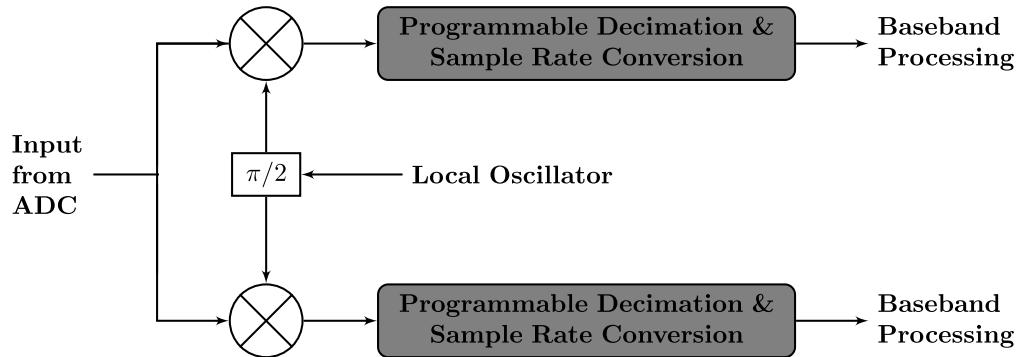
SDR receivers must be capable of extracting non-uniform bandwidth channels with minimum reconfiguration overhead. Among the channelization methods presented above, PC and FRMFB are the two approaches suitable for non-uniform bandwidth channelization. In FRMFB approach, it is required to redesign four filters with the change in radio communication standard, leading to offline computations and high reconfiguration overhead. The hardware complexity of PC approach is slightly high when compared to FRMFB approach, but its reconfiguration overhead is low as required for SDR receivers. In PC approach, it is required to redesign one channel filter unlike four filters in FRMFB approach to support band pass channel of a new radio communication standard. The computational complexity of a channelizer in SDR receiver is very high due to its operation at high IF sampling rates. This problem can be addressed by preceding the channelizer with digital down converter(DDC) and sample rate conversion (SRC) filters.

### 3.2 Digital Down Conversion-Sample Rate Conversion

Among the various channelization approaches presented in Section 3.1, it is observed that the PC approach is suitable for non-uniform channelization with minimum reconfiguration overhead. In addition, the channel filter and SRC filter for channel extraction can be merged into a single filter to reduce hardware complexity.

Digitization of wideband IF signal at Nyquist rate leads to oversampling of narrow band radio channels due to the phenomenon of band pass sampling. The required channel of interest from the wideband IF signal can be extracted by employing band pass filters. Band pass filtering performed at oversampled rate of radio channel leads to high

computational complexity. This can be addressed by decimating the sampling rate of the required channel of interest by its oversampling ratio.



**Figure 3.7** Digital IF Stage Architecture

The process of sampling rate alteration employs anti-aliasing and anti-imaging low pass filters for decimation and interpolation, respectively. Hence, to reduce the sample rate of channel of interest, the band pass channel needs to be converted into a low pass channel, i.e., the centre frequency of the channel has to be shifted from a non-zero frequency to zero or DC frequency.

Fig.3.7 shows the architecture of digital IF stage consisting of digital quadrature mixer followed by a programmable digital decimator and a sample rate converter. The functionality of the architecture is described as follows. The composite wideband digital IF signal (Input samples from ADC) is mixed in a quadrature mixer with a frequency equal to the centre frequency of the channel of interest. The anti-aliasing low pass filters employed in the process of SRC, extracts the low frequency information signal while attenuating the high frequency carrier signal components.

### 3.2.1 Quadrature Mixers

The implementation of DDC requires quadrature waveforms to be generated. There are two methods for generation of quadrature waveforms viz., look up table method and Coordinate Rotation DIgital Computer (CORDIC) method.

In look up table method, ROM of size  $MXN$  is used, where  $M$  represents the number of bits used to represent phase resolution and  $N$  represents the amplitude resolution. The

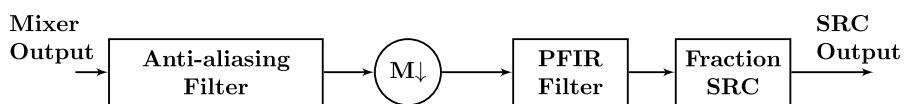
locations of the ROM have to be filled with the required cosine/sine values and they are accessed with the phase address ( $M$ ). The disadvantages of this method are exponential increase in the size of ROM with increase in phase/amplitude resolution; a phase shifter introduces phase distortions for the generation of quadrature waveform; and it requires explicit multipliers for mixing. These problems are addressed by implementing direct digital synthesizers (DDS) based on CORDIC architectures.

The COordinate Rotation DIgital Computer (CORDIC) is a fast technique, first developed by J.E. Volder in 1959, for computing trigonometric functions using shift and add operations and conversion from rectangular to polar conversion [18], [19]. CORDIC algorithm can be realized using bit serial architecture and word parallel/pipelined architecture. The choice of architecture depends on the required throughput, frequency of operation of a target application and area constraints [20].

### 3.2.2 Programmable Sample Rate Conversion

A programmable SRC decimates the sampling rate of the narrowband radio channel from IF sampling rate to a suitable sampling rate such that the signal processing carried out in the subsequent stages of SDR receiver can be implemented with reduced computational complexity using digital signal processors (DSPs) or reconfigurable hardware or software.

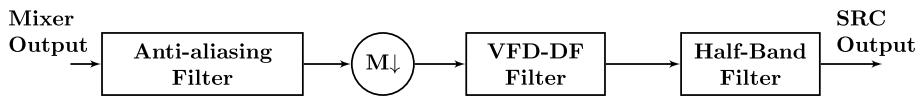
In SDR receivers, the baseband processing has to be carried out at a rate much lower than the IF sampling rate. The required SRC factor is selected such that the output sampling rate is an integer multiple of Nyquist sampling rate of baseband signal. In such a case, the required SRC factor may be an integer or a rational factor.



**Figure 3.8** Conventional SRC Architecture

Fig.3.8 shows the architecture of a conventional programmable sample rate decimator to achieve SRC by rational factors. It consists of integer rate decimator followed by a

programmable FIR (PFIR) filter and arbitrary SRC. The integer rate decimation factor is selected in a way such that it can elevate the baseband signal of the required channel of interest and the adjacent radio channels in the wideband signal are suppressed. Since, the sampling rate is not an integral multiple of channel spacing, a PFIR filter is employed to remove the adjacent channel interference (ACI). Finally, an arbitrary SRC converts the sampling rate of the signal suitable for baseband processing which is usually an integer multiple of sampling rate of baseband signal. This architecture can be employed to extract radio channels of variable bandwidths [21]. However, it may be noted that the hardware and the computational complexity of the conventional architecture is high. This is due to the requirement of high speed general purpose multipliers for the implementation of PFIR filters and the requirement of an interpolator and decimator to achieve SRC by fractional rates.



**Figure 3.9** Variable Digital Filter based SRC Architecture

Fig.3.9 shows an efficient architecture employing variable fractional delay digital filters (VFD-DF) and half band filters (HBF) [22]. VFD-DF eliminates the need of interpolation and decimation to achieve fractional SRC and the high complexity PFIRs are replaced by the fixed coefficient HBFs to reduce ACI. The overall SRC achieved employing this architecture can be expressed as:

$$M_{SRC} = R * \mu * 2^K \quad (3.3)$$

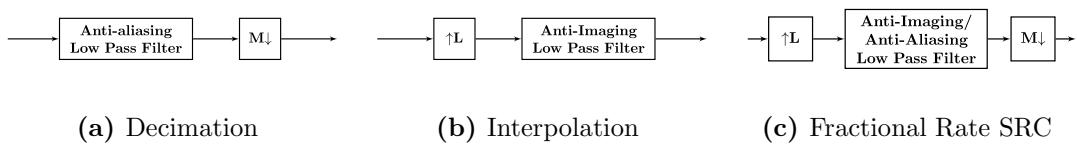
where,  $M_{SRC}$  = Required SRC factor,  $R$  = Integer rate decimation factor,  $\mu$  = Fractional SRC factor, and  $K$ =Number of half-band filters employed.

For SDR receivers, the filters in the SRC filter architecture must be reconfigurable with minimum reconfiguration overhead to support multi-standard radio communications while achieving low hardware complexity and computational complexity. Among the various filters in SRC filter chain, the coefficients of HBF filter are fixed with a fixed pass band edge. Hence, the focus of our research work is to design a reconfigurable integer

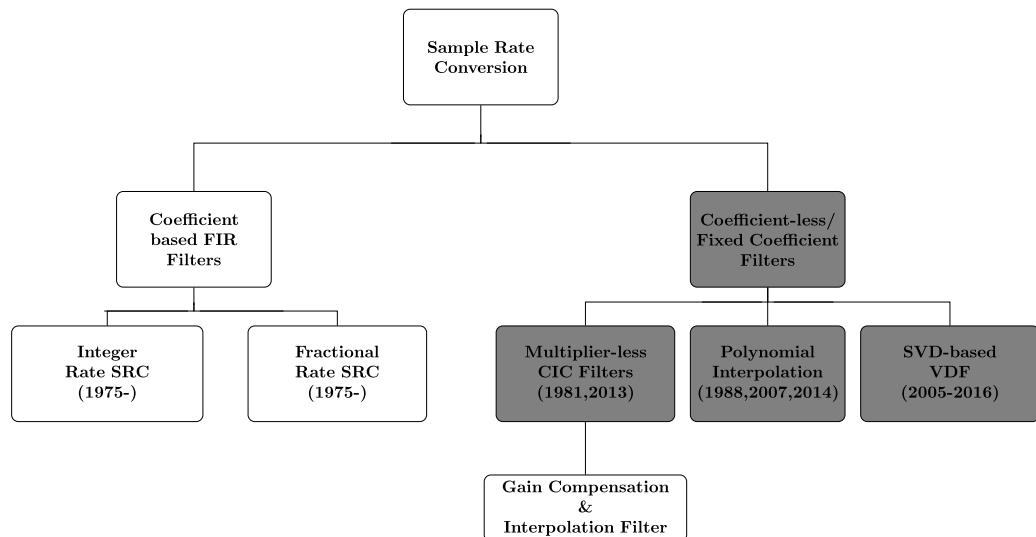
rate decimation filter and VFD filter in the SRC architecture (See Fig.3.9) and various approaches for the design of multi-rate SRC filters are presented in the next section.

### 3.3 Multi-Rate SRC filters

Design of multi-rate filters proposed in the literature for achieving SRC and their suitability to employ in SDR receivers is presented in this section.

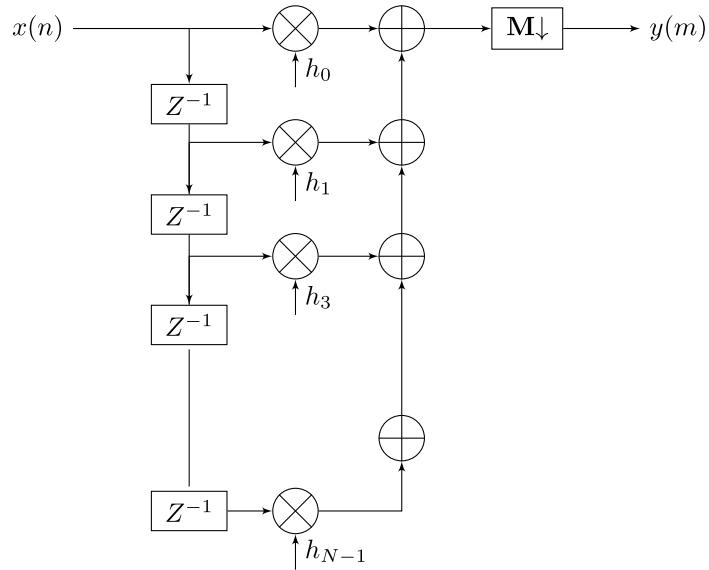


**Figure 3.10** Sample Rate Conversion structures



**Figure 3.11** Sample Rate Conversion Architectures proposed in the literature

Fig.3.10a, 3.10b and 3.10c shows the structures for achieving integer rate decimation, interpolation and fractional rate interpolation, respectively. It is observed that SRC structures require anti-aliasing low pass filter for decimation and anti-imaging low pass filter for interpolation. Fig.3.11 shows various methods to design these filters with an

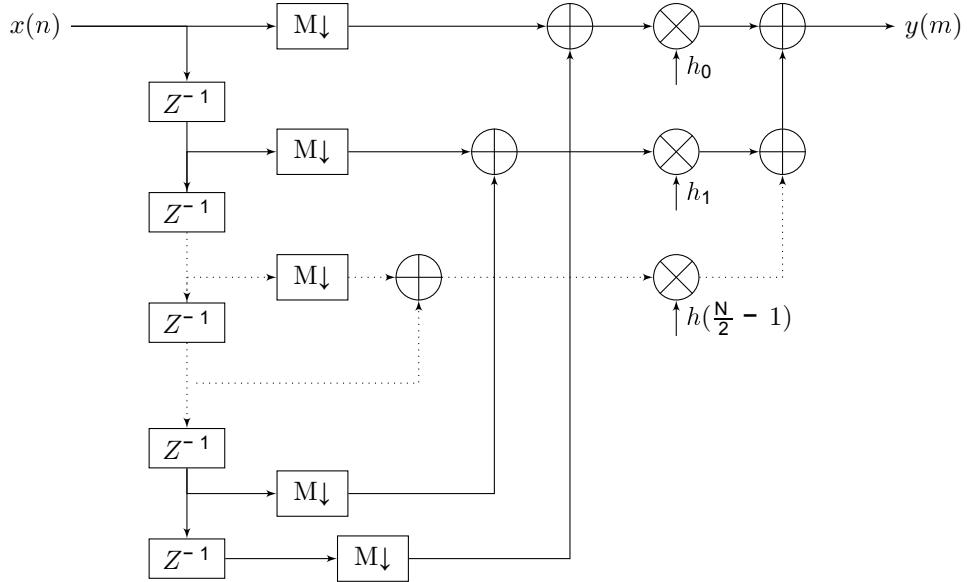


**Figure 3.12** Direct Form FIR Structure for Decimation

emphasis to reduce computational complexity and achieve reconfigurability as required in SDR receivers.

### 3.3.1 FIR filter based SRC structures

The structure of decimation filter employing direct form FIR structure is shown in Fig.3.12. FIR filters for SRC can be designed to meet the required signal specifications by employing various FIR filter design methods proposed in the literature [23]- [26]. Consider the original sampling rate of the signal as  $F$  and the new sampling rate for decimation by a factor  $M$  as  $\frac{F}{M}$ . It is observed from the structure of decimator (See Fig.3.12) that the anti-aliasing low pass filters operates on  $M$  samples of the input signal and discards  $M - 1$  samples of the filtered signal. This results in high computational complexity. If the FIR filters are designed such that the filter coefficients are symmetric in nature, the number of filter computations can be approximately reduced to half [27]. The computational complexity can be further reduced by applying the principle of duality and transposition [28]- [29]. By exploiting these properties, the direct form FIR structure in Fig.3.12 can be transformed into a structure shown in Fig.3.13 with the number of computations reduced to half and the filter output is computed at a lower sampling rate of  $\frac{F}{M}$ . The implementation of FIR filters employing poly-phase structures further reduces



**Figure 3.13** Application of Duality and Transposition in FIR filter based SRC Architecture

the hardware complexity. This is due to the implementation of poly-phase structures employing fast convolution methods based on Fast Fourier Transform techniques [30]. A time-varying structure for fractional rate interpolation employing poly-phase structures is presented. The filter employs a poly-phase interpolator with a down sampler or a poly-phase decimator with an up sampler. This leads to high computational complexity as it requires design of both an interpolator and decimator. A special class of filters called half-band filters can be employed to achieve SRC by a factor of two. Since, half of the filter coefficients in HBF are zero, it may be observed that its hardware complexity is reduced to half [31]- [34]. For the design of FIR filters, the order of FIR filter and the computational complexity can be approximately expressed as

$$N \approx K * \frac{F}{F_s - F_p} \quad (3.4)$$

$$CC = \frac{N * F}{2 * M} \quad (3.5)$$

where,  $K$  = Constant dependent on pass band and stop band ripple,  $F$  = Sampling rate of the signal,  $F_s - F_p$  = Transition bandwidth of the filter,  $F_s$  = Stop band edge frequency,  $F_p$  = Pass band edge frequency,  $CC$  = Computational complexity and  $M$  = SRC factor.

It may be observed from Eq. 3.4 that the order of the filter is directly proportional to the sampling frequency of the signal and inversely proportional to the transition band-

**Table 3.2** Hardware and Computational Complexity of FIR Filter based SRC Structures proposed in the literature

Filter Structure	SRC Factor	Filter Order	# MACs	Comp.Complexity
Direct Form FIR [25]	$M$	$N$	$N$	$NF$
Symmetric FIR [27]	$M$	$N$	$\frac{N}{2}$	$\frac{NF}{2}$
Transposed SRC [29]	$M$	$N$	$\frac{N}{2}$	$\frac{NF}{2M}$
Half-Band FIR [34]	2	$N$	$\frac{N}{4}$	$\frac{NF}{4}$
Multi-stage SRC [36]	$M = \prod_{i=1}^I M_i$	$\sum_{i=1}^I N_i$	$\sum_{i=1}^I \frac{N_i}{2}$	$\sum_{i=1}^I \frac{N_i}{2} * \frac{F}{\prod_{j=1}^i M_j}$

width of the signal. Hence, the design of single stage FIR filter with large SRC factor results in higher order filter. It can be inferred from Eq.3.5 that the higher order filter at high sampling frequency leads to high computational complexity. Multi-stage implementation reduces the filter design complexity by selecting wider transition bandwidth in the initial stages and sample rate reduction in its later stages of SRC filter design [35]- [37]. Low order filters results in reduction in computation complexity, storage complexity and finite word-length effects such as round-off noise, coefficient sensitivity.

Table 3.2 summarizes the computational complexity of SRC filter designed based on FIR filter design methods. It may be noted that coefficient based FIR filters are suitable to achieve SRC by fixed factors. As the required signal characteristics vary with varying SRC factors, the filter needs to be redesigned which involves high reconfiguration overhead, an undesirable feature for SDR receivers. It may also be noted that the computational complexity of SRC filter is directly proportional to the input sampling rate of the signal. In SDR receivers, coefficient based SRC filter requires operation of multipliers at wideband IF sampling rates, leading to high computational complexity and high power dissipation. These problems can be addressed by employing coefficient-less and fixed coefficient architectures in the design of SRC filters for SDR receivers.

### 3.3.2 Coefficient-less/Fixed Coefficient SRC structures

#### Multiplier-less SRC filter structure

In 1974, Peled and Liu [38] has proposed a coefficient slicing approach by employing read-only-memories (ROMs) and adders for the realization of fixed coefficient digital filters. In the design of decimation/interpolation filter, the aliasing or the imaging errors must be limited. In 1977, multiplier-less architecture for sample rate change by a factor of two has been proposed by Goodman and Carey [39]. As it is required to attain SRC by large factors in SDR receivers, multiple stages of such filters are required.

In 1981, E.B. Hoganeur [40] has presented an economic class of digital filters called cascaded-integrator-comb (CIC) filters which employ digital integrator and comb sections to meet the design requirements of decimation and interpolation filters. CIC filters are efficient in terms of computational complexity, easily reconfigurable and require limited storage due to their multiplier-less architectures. High bit growth rate of registers due to high gain of integrators, gain droop in the pass band of interest and their unsuitability to attain fractional rate SRC are the drawbacks of CIC filter. High bit growth rate requirement can be relaxed by normalizing the gain at the output of each integrator to unity. The restoration of gain droop in the pass band and fractional rate SRC can be achieved by cascading a CIC compensation filter and fractional rate SRC filter with CIC filter, respectively. Since these filters operate at a very low sampling rate when compared to the CIC filter, a significant reduction in the computational complexity is observed in the design of SRC filters.

#### Polynomial interpolation based SRC Filter Structure

In 1984, T.A. Ramstad [41] has proposed a method based on analog and Lagrange interpolator to achieve SRC among arbitrary sampling rate frequencies. It requires online computation of coefficients with high computation rate to adapt to the changes in the SRC factor. In 1988, C.W. Farrow [42] has proposed a filter structure with fixed coefficients to achieve SRC by fractional rates. The Farrow filter consists of sub-filters with fixed coefficients and a fractional rate interpolating structure. A linear time invariant structure,

derived by employing numerical interpolation techniques such as Lagrange interpolation and implemented using Farrow structure is presented [43]. The fixed coefficients of the sub-filters are implemented as SOPOT coefficients to reduce hardware complexity [44]. F.M. Gardener et.al presents the required fundamentals for interpolation and implementation aspects in digital modems [45], [46]. It is concluded that the Farrow structure is an efficient structure in terms of implementation as it requires only one control parameter and employs lesser multiplications [46]. A modified Farrow structure employing a transformation matrix to reduce the number of additions and multiplications is proposed [47], [48], by eliminating the integer part of the rational SRC factor. A method to reduce computational complexity by employing time-varying CIC structures is proposed in [49]. An efficient structure for Lagrange interpolation is proposed by Canadan [50]. In this structure, the filter is designed based on Taylors series expansion and the order of the filter can be varied online but results in high hardware complexity due to their multiplier based architectures. The drawback of SRC filters designed employing polynomial interpolation method is that they are designed based on time domain interpolation polynomials without considering the spectral characterisitics of the signal.

### **SVD-VDF based SRC Filter structure**

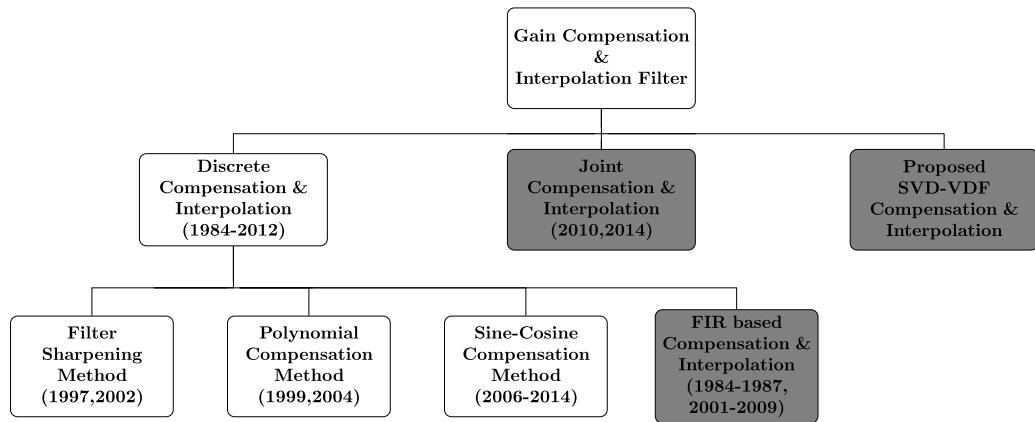
In 1989, a frequency domain based variable digital filter(VDF) is proposed to compute filter coefficients online and attain variable spectral characteristics such as pass band edge frequency and variable fractional delay [51]. The architecture of VDF employs fixed coefficient filter cascaded with multi-dimensional polynomials expressed in terms of variable spectral parameters. The coefficients of multi-variable polynomials are computed by employing Gill and Murrays algorithm [52]. In 1994, Tian-Bo Deng has proposed a VDF based on outer-product expansion method [53]. In this technique, a multi-dimensional array based on 1D magnitude response is constructed and then outer product expansion method is employed for expanding the multi-dimensional array as the sum of outer products of 1D arrays (vectors). This results in reducing the problem to design of constant coefficient 1D digital filters and 1D polynomial approximations. This method is extended to design 1D and 2D VDF with arbitrary magnitude response and improves the accuracy by minimizing the weighted error between the desired and actual frequency response in

the least squares sense [54], [55], [56]. The design accuracy of the VDFs presented so far depends on the grid density of the spectral parameters with which they are discretized.

An optimum closed form solution, independent of sampling grid densities and discretization free method to design variable fractional delay (VFD) filter is presented [57]. In this technique, the objective error function of variable frequency response is minimized by employing numerical integration technique which increases the design accuracy and reduces the computational complexity. Based on this, a closed form solution for the design of variable 1D digital filter with simultaneously tuneable magnitude and tuneable fractional phase-delay responses is presented [58]. In this technique, the coefficients of a variable FIR is expressed as a 2D polynomial in terms of a pair of parameters called spectral parameters; one is for independently tuning the cut-off frequency of the magnitude response, and the other is for independently tuning fractional phase-delay.

A straight forward method to design variable digital low pass filters with variable fractional delay (VFD) based on singular value decomposition (SVD) of the desired variable frequency response is presented in [59]. SVD technique generates complex vectors and real vectors. The complex vectors are considered as the frequency responses of the 1-D constant FIR filters designed with symmetrical or anti-symmetrical coefficients to reduce computational complexity. Similarly, real vectors can be regarded as the desired values of 1-D polynomials with either even symmetry or odd symmetry. The SVD-VDF based VFD filter with one variable parameter is extended to design a VDF with multiple variable spectral parameters. In this method, the original 1D-variable filter is decomposed using SVD algorithm and implemented using constant coefficients 1D-FIR filters and multi-dimensional polynomial approximations obtained by constant vector-array decomposition (VAD) [60], [61].

Hence, it can be concluded that the coefficient-less CIC structures reduces the complexity involved in the design of reconfigurable SRC filters required for SDR receivers. However, it is required to cascade a gain droop compensation filter to attain the required spectral characteristics and a fractional rate interpolation filter to achieve the required symbol rate of a radio standard.



**Figure 3.14** Gain Compensation and Interpolation methods Proposed in the literature

### 3.4 Gain droop compensation and Interpolation filters

This section presents the methods to design CIC compensation and fractional rate interpolation filters to restore the gain droop and achieve the required symbol rate. A brief review of various methods proposed in the literature to design these filters is presented in Fig.3.14.

#### 3.4.1 Discrete Compensation and Interpolation filter

CIC gain droop compensation filter based on the required signal characteristics can be designed as a single stage FIR filter employing FIR filter design methods presented in [62]- [64]. The hardware complexity depends on the order of the FIR filter which is a function of transition bandwidth. (See.Eq.3.4).

In 1997, the authors have proposed a filter sharpening method by employing multiple copies of the same CIC filter and half band filters to restore the gain droop in the pass band [65]- [67]. In these architectures, multiple copies of CIC filters operate at high input sampling rates. This implementation rate can be reduced from high input sampling rate to lower sampling rates by employing poly-phase decomposition technique [68] and two stage implementation technique [69].

An area efficient low power droop correction filter employing half-band filters is presented [70]. The pass band and stop band edge frequencies are fixed in the design of

half-band filters. Since transition band of half band filters is fixed this method cannot be employed to support multi-standard radio communications and requires complex PFIR filters to support variable transition bandwidths of multiple radio standards. Hyuk et.al [71] have proposed a method to reduce the complexity of the half-band and PFIR filters by cascading the output of the CIC filter with a second order CIC compensator with the transfer function,

$$H_{comp}(z) = a + bz^{-1} + az^{-2} \quad (3.6)$$

The filter coefficients  $a, b$  in Eq. 3.6 are function of required signal characteristics and they can be computed by employing conventional filter design methods [72]. The implementation complexity of this filter can be reduced significantly by expressing the fixed coefficients  $a$  and  $b$  as sum of powers of two (SOPOT) coefficients or Canonic signed digits (CSD) [73], [74]. However, these coefficients vary with the change in the specification of radio communications standard.

In 2006, Kim et al have suggested a simple method for the design of CIC compensation filter for WCDMA/CDMA2000 receiver. A low complexity three coefficient symmetric filter is designed to achieve the required compensation in order to reduce hard-wired coefficients [75]. The three coefficients are chosen as  $[\frac{-a}{(1-2a)}, \frac{1}{(1-2a)}, \frac{-a}{(1-2a)}]$  and its frequency response is expressed as:

$$H_{comp}(\omega) = \frac{(1 - 2a\cos(\omega))}{(1 - 2a)}, \quad a \neq 0.5 \quad (3.7)$$

Coefficient  $a$  in Eq.3.7 is computed by minimizing the mean square error between the ideal desired response and actual response of the filter. The limitation of this method is the input sampling rate must be an integral multiple of the required symbol rate and its spectral response is dependent on the value of  $a$ .

A new two stage based CIC compensator which eliminates the need of computation of coefficients is presented in [76]. In the first stage of compensator, a sine filter with fixed integer coefficient is employed to restore the gain droop in pass band which in turn increases the side lobes in the stop band. These side lobes can be reduced by employing a cosine filter in the subsequent stage. Since the sine and cosine filters are filters with fixed integer coefficients, they can be implemented by employing simple shift and add operations. It may be observed that increase in number of stages of sine and cosine filter

improves the frequency response in pass band and stop band, respectively with a slight increase in computational complexity.

A simple method for design of narrow band and wide band compensation filters based on sine/cosine based compensators has been proposed by G.J. Dolecek [77]- [78]. This approach presents the design of compensator with only one parameter, the order of the CIC filter,  $N_{CIC}$ . The frequency response and the system transfer function of sine based compensator with  $R$  as CIC decimation factor are expressed as:

$$H_{comp}(\omega) = |1 + 2^{(b+2)} \sin^2\left(\frac{\omega R}{2}\right)| \quad (3.8)$$

$$H_{comp}(z) = -2^{-(b+2)}(1 - (2^{(b+2)} + 2)z^{-M} + z^{-2M}) \quad (3.9)$$

where,  $b$  is a parameter dependent on  $N_{CIC}$  and varies with wide band and narrow band compensation. The hardware realization of this approach requires three adders. The gain droop in the pass band is improved by sharpening the frequency response of CIC filters as indicated in [65] and cascading it with a sine-based compensator [79].

A closed form compensator based on maximally flat error condition is proposed in [80]. The bandwidth of this filter increases with increase in the number of coefficients in the filter. It is observed that this filter provides better frequency response at the expense of more multipliers when compared with the other methods presented in the literature [65]- [79].

A digital down converter chip, GC4016, based on inverse frequency response of the CIC filter has been designed and launched by Texas Instruments in 2009 [81]. In this DDC chip, the filter coefficients of the compensation filter are programmed through microprocessor based register programming to attain the required signal characteristics.

A unified structure for providing both narrowband and wideband compensation is presented in [82]. This compensation filter can be a second order or fourth order filter with symmetric coefficients. The derived coefficients are function of decimation factor,  $R$ , number of stages of CIC filter  $N_{CIC}$  and a parameter describing wideband or narrowband bandwidth. This compensation filter has employed canonical signed digit multipliers with minimum number of adders to achieve a multiplier-less architecture. This compensator

can be designed as narrow band and wide band compensators with bandwidth of  $0.0078R\pi$  and  $0.0156R\pi$ , respectively. CIC gain droop of  $-1.12\text{dB}$  is reduced to  $-0.12\text{dB}$  for narrow band compensator and  $-4.54\text{dB}$  to  $-0.58\text{dB}$  for wideband compensator.

The compensator with SOPOT coefficients based on interval analysis and mini-max error criterion is proposed [83]. This filter is designed with odd number of coefficients,  $M_{coeff}$  and  $P$  terms for computing SOPOT coefficient with a pass band edge of  $0.25\pi$ ,  $0.5\pi$  and  $0.6\pi$ . It may be observed that this exhibits a better compensation with low computational complexity. However, the number of SOPOT coefficients vary with required pass band edge and thereby the computational complexity of the filter.

**Table 3.3** Hardware and Computational Complexity of CIC Compensation Filters proposed in the literature

Design Method	Multipliers	Adders	Operating Frequency
Filter Sharpening [65]	2	$3N_{CIC}$	$F$
Filter Sharpening with HBFs [66], [67]	$2, \frac{N}{2}$	$3N_{CIC}, \frac{N}{2}$	$F, \frac{F}{R}$
Polyphase CIC & Filter Sharpening [68]	2	$3N_{CIC}$	$\frac{F}{2}$
Two-Stage CIC & Filter Sharpening [69]	2	$3N_{CIC}$	$\frac{2F}{R}$
Polynomial Compensation [71], [74]	2	2	$\frac{F}{R}$
Flat Pass band [75]	3	2	$\frac{F}{R}$
Two-stage Sine-Cosine Compensator [76]	—	6	$\frac{F}{R}$
Sine/Cosine Compensator [77]- [78]	—	3	$\frac{F}{R}$
Sine Compensation & Sharpening [79]	2	$(3N_{CIC}, 2)$	$(\frac{2F}{R}, \frac{F}{R})$
Closed form Compensation [80], [81]	$N_{comp}$	$N_{comp}$	$\frac{F}{R}$
Max. Flat Pass band, $N = (2, 4)$ [82]	$(3, 5)$	$(3, 5)$	$\frac{F}{R}$
Interval Analysis [83]	—	$M * P$	$\frac{F}{R}$

Table 3.3 summarizes the hardware complexity in terms of number of multipliers and adders and the frequency of operation of the various CIC gain droop compensation filters presented in the literature. It may be noted that the hardware complexity of the compensation filters is low with reduced number of multipliers operating at a very low sampling rate of  $\frac{F}{R}$  with fixed pass band edge and stop band edge. Hence, this compensation FIR filters are cascaded with a programmable FIR filter to attain the

required spectral characteristics of a specific radio standard into consideration as described in [66], [67], [80], [81]. Further, it is required to cascade a fractional rate interpolation filter to make sample rate of the signal power of two multiple of baseband signal sampling rate. It is observed that the independent design of integer rate SRC and fractional rate SRC filters degrades the overall frequency response of the required filter [84].

### 3.4.2 Joint compensation and interpolation filter

Faheem Sheikh has proposed an efficient architecture for sample rate conversion in software radio receivers [85]. In this architecture, the required rational SRC is achieved by employing multiplexed CIC filters and Farrow structures to achieve integer rate SRC and fractional rate SRC, respectively. Since it is required to compensate the gain droop of CIC filters and make the sample rate of signal equal to the multiple of baseband symbol rate, it is required to cascade a gain droop compensation filter and a fractional rate interpolation filter. A joint compensation and interpolation filter is designed based on the filter with the required frequency response expressed in terms of frequency domain polynomials and implemented using Farrow structure [86], [87]. The coefficients of the sub-filters in the Farrow structure are derived such that the error between the desired frequency response and the actual frequency response is minimized either in the least squares or mini-max sense.

## 3.5 Conclusions

In this chapter, a brief survey of various techniques proposed in the literature to implement the process of channelization with reduced hardware and computational complexity is presented. It is observed that the design of multi-standard radio receivers can employ PC approach to extract the required radio channel from wideband digital IF signal as its hardware implementation requires less reconfiguration overhead to support any existing or forthcoming radio standard. It is also observed that the computational complexity involved in the implementation of the channelization process can be reduced by performing digital down conversion and sample rate conversion before channelization. The hardware complexity and the reconfigurability of various techniques available in the

literature to design reconfigurable SRC filter are compared. It is observed that the design of SRC filter employing coefficient-less CIC filter results in low hardware complexity and low reconfiguration overhead. However, it may be noted that the frequency response of CIC filters exhibits gain droop in the pass band and support SRC by integer rates. Hence, it is required to cascade a gain droop compensation and fractional rate interpolation filter. The performance of various method for the design of gain droop compensation and interpolation filters proposed in the literature are compared. It is observed that the joint compensation and filter does not require programmable FIR and fractional rate SRC filter resulting in low hardware complexity.

In this work, we are interested in designing reconfigurable SRC filter with minimum reconfiguration overhead and better spectral characteristics for extracting channels of multiple radio communication standards from the wideband digital IF signal. Design of SRC filter for four radio communication standards with low hardware complexity and improved spectral characteristics is presented in the next chapter.

# Chapter 4

## Sample Rate Conversion Filter for SDR Receivers

This chapter presents the design of Sample Rate Conversion (SRC) filter by employing cascaded-integrator-comb (CIC) filter and Farrow structures available in the literature. The required SRC is achieved by employing CIC filters in cascade with CIC gain compensation and interpolation filter. The functionality of the designed multi-stage CIC filters is verified by implementing on field programmable gate array (FPGA) and the performance parameters such as frequency of operation and the hardware complexity of gain optimized CIC filter is compared with the non-optimized CIC filter. The designed CIC filter is simulated in MATLAB to observe its frequency response. Since CIC filter results in gain droop in the pass band, a modified joint compensation and interpolation filter is designed to compensate the CIC gain droop and achieve interpolation by fractional rates. In addition, VHDL model for modified joint compensation and interpolation filter is developed and implemented on Kintex-7 FPGA to compare the hardware complexity with the existing compensation and interpolation filters available in the literature.

### 4.1 Introduction

In SDR receivers, the incoming RF signal from the antenna is translated into wideband IF signal using low noise amplifiers and low pass filters and implemented with fixed analog hardware. The wideband IF signal which incorporates narrow band channels of multiple radio standards is digitized with fixed ADC at Nyquist rate. Then the process of channelization and baseband processing is carried out. The process of channelization

extracts the required radio channel at the symbol rate and spectral characteristics as specified by the specifications of the radio standard into consideration. Digitization of wideband IF signal oversamples the narrowband radio channels due to the phenomenon of pass band sampling. This in turn leads to high computational complexity and high power dissipation for the process of channelization, an undesirable feature for battery operated hand held devices. Hence, a sample rate conversion (SRC) filter interoperable between the IF sampling rate and symbol rate of the radio standard has to be designed to overcome these limitations. For SDR receivers, an SRC filter with programmable SRC factors and reconfigurable spectral characteristics with low computational complexity and low power dissipation is required to support multi-standard radio communications.

## 4.2 Reconfigurable Architecture for Sample Rate Conversion Filter

In this section, the specifications derived for our work to design and implement a reconfigurable SRC filter architecture for supporting multi-standard radio communications are presented.

### 4.2.1 Design specifications

SRC filter in the IF stage of software radio receiver is designed and analyzed by considering four wireless communication standards, namely WiMAX (IEEE 802.16), WCDMA (UMTS), CDMA2K (IS-95) and GSM900. The single channel bandwidths and the corresponding data rates of these radio standards considered are enumerated in Table 4.1. In this work, we have considered an 80MHz wideband IF signal for receiving signals of four radio standards and sampled at Nyquist rate of 160MSPs. The narrowband radio channels present in the wideband IF signal get oversampled due to the phenomenon of pass band sampling. As the baseband operations such as demodulation, decoding, equalization etc., are performed at standard specific data rate, it is required to process the oversampled radio channel by an SRC stage. The required SRC factors expressed as the ratio of IF sampling rate to the sampling rate of single channel bandwidth, are presented

---

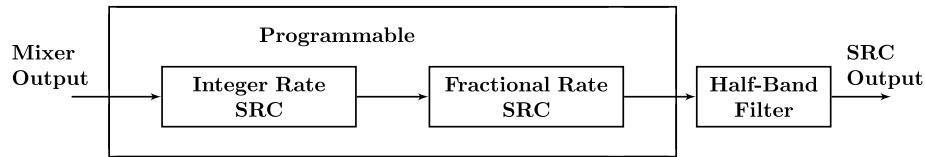
**Table 4.1** Required SRC Ratio to support Multi-standard Radio Communications

Radio Standard	WiMAX 802.16	WCDMA (UMTS)	CDMA-2000	GSM-900
IF Bandwidth (in MHz)	80	80	80	80
IF Sampling Rate (in MSPs)	160	160	160	160
Single Channel Bandwidth (in MHz)	10	5	1.25	0.2
Required Sampling Rate (in MSPs)	20	10	2.5	0.4
Oversampling Ratio (OSR)	8	16	64	400
Symbol Rate (in Mbps)	10	3.84	1.2288	0.270833
SRC Ratio	$\frac{1}{16}$	$\frac{3}{125}$	$\frac{24}{3125}$	$\frac{677}{400000}$

(See Row 6 in Table 4.1). It may be noted from the specifications that it is required to design a single structure for SRC filter with variable integer rate SRC and fractional rate SRC factors to accommodate multi-standard radio signals.

#### 4.2.2 Programmable SRC Architecture for SDR receivers

Figure 4.1 shows the conventional architecture available in the literature for programmable SRC to be used in multi-standard SDR receivers. This architecture consists of a cascade of programmable integer rate SRC filter, programmable fractional rate SRC filter and k-stages of half band filter.



**Figure 4.1** Block Diagram of Programmable Sample Rate Conversion Filter for SDR Receivers

The programmable integer rate SRC (ISRC) filter is employed to achieve decimation by large integer factors. The programmable fractional rate SRC filter (FSRC) is used to interpolate the sampling rate of the signal as the power-of- two multiple of symbol rate. Half band filters are fixed coefficient filters employed to achieve the required symbol rate.

**Table 4.2** Computation of SRC Factors to support Multi-standard Radio Communications

Radio Standard	WiMAX 802.16	WCDMA (UMTS)	CDMA-2000	GSM-900
SRC Ratio, $\frac{p}{q}$	$\frac{1}{16}$	$\frac{3}{125}$	$\frac{24}{3125}$	$\frac{677}{400000}$
Integer Rate Decimation, $R$	8	16	64	384
Fractional Rate Decimation, $\mu$	1	1.536	1.966	1.299
Number of HBFs, $k$	1	2	2	1

The required SRC ratio is attained by employing this factorizing scheme and expressed in terms of integer rate decimation factor, fractional rate interpolation factor and the number of half band filters is given by Eq.4.1 [85].

$$\frac{p}{q} = \frac{\mu}{R * 2^k} \quad (4.1)$$

where,  $R$ =Programmable integer rate decimation factor,  $\mu$ = Programmable fractional rate interpolation factor, and  $k$  = Number of half-band filter stages.

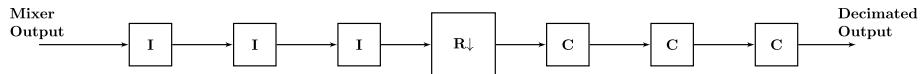
Design of programmable SRC filter for four radio standards involves the selection of appropriate values for integer rate decimation factor ( $R$ ), the fractional rate interpolation factor ( $\mu$ ), and the number of half-band filter stages ( $k$ ) to achieve required SRC ratio. These values are computed using Eq.4.1 and presented in Table 4.2. Since the half-band filters are fixed coefficient filters with a decimation factor of two in each stage, the design of integer and fractional rate SRC filters is only presented in this work. The programmable ISRC is implemented by employing coefficient-less CIC filter and the fractional rate SRC filter is implemented as a fixed coefficient polynomial interpolation filter with variable SRC factors as stated in Chapter 3. These filters are easily reconfigurable due to their coefficient-less or fixed coefficient structures. The next section presents the design of CIC filters to support wideband IF signal with low computational complexity and power dissipation.

### 4.3 Integer Rate Sample Rate Conversion

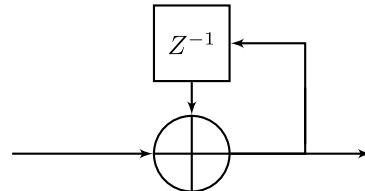
In this section, the design and FPGA implementation of different CIC filters to achieve SRC by large integer factors and to support wideband IF signals is presented.

#### 4.3.1 CIC Filter Design

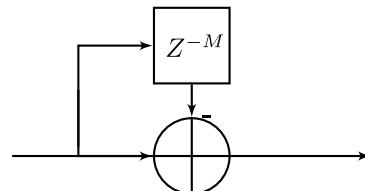
In 1981, E.B. Hoganeur [40] has proposed a multiplier-less architecture to achieve SRC by large integer rates. Fig.4.2 shows the structure of CIC filter consisting of integrator, decimator and comb filter. An  $N^{th}$  order CIC filter consists of  $N$  integrators, decimator and  $N$  combs. The integrators in the CIC filter implements the functionality of an anti-aliasing low pass filter while the comb structure eliminates the effect of redundant samples in the signal. The transfer function of CIC filter is expressed as cascaded transfer



(a) Structure of CIC Filter



(b) Integrator Structure



(c) Comb Structure

**Figure 4.2** Sample Rate Conversion by Large Integer Rates

function of  $N$  integrators and  $N$  combs is given as:

$$H_{CIC}(Z) = H_I^N(Z) * H_C^N(Z) \quad (4.2)$$

It is observed that, if the integrators in CIC filter operates at a frequency of  $F_s$  and  $R$  is the decimation factor, then the comb structures operate at a frequency of  $\frac{F_s}{R}$ . Hence, the transfer function of one integrator and one comb structure with  $M$  differential delay elements is expressed as:

$$H_I(Z) = \frac{1}{1 - Z^{-1}}, \quad H_C(Z) = 1 - Z^{-RM} \quad (4.3)$$

The transfer function of CIC filter obtained by substituting Eq. 4.3 in Eq. 4.2 is indicated as:

$$H_{CIC}(Z) = \left( \frac{1 - Z^{-RM}}{1 - Z^{-1}} \right) \quad (4.4)$$

With  $Z = e^{j2\pi f}$ , the magnitude response of the CIC filter is given by

$$H_{CIC}(f) = \left| \frac{\sin(\pi Mf)}{\sin(\frac{\pi f}{R})} \right|^N \quad (4.5)$$

For large decimation factors as required in SDR receivers Eq.4.5 can be approximated as:

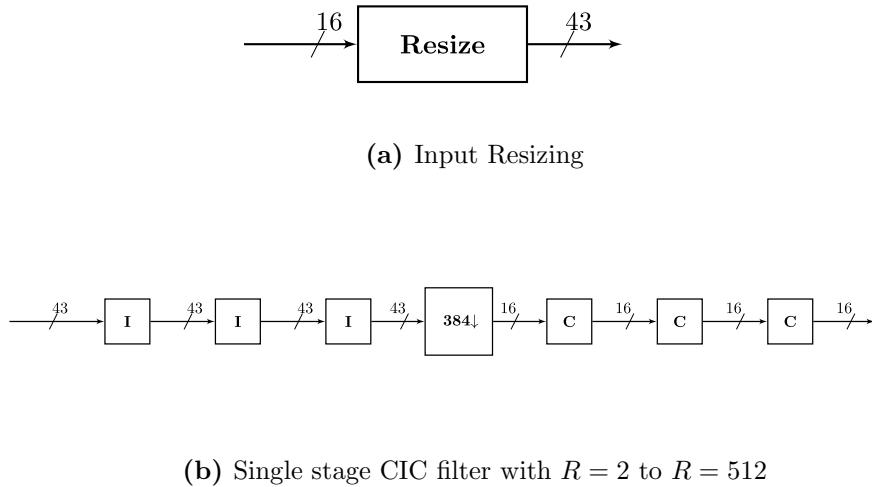
$$H_{CIC}(f) = \left| RM \frac{\sin(\pi Mf)}{(\pi Mf)} \right|^N \quad (4.6)$$

It may be noted from Eq.4.6 that the frequency response of CIC filter is sinc in nature and the gain of CIC filter increases with increase in value of decimation factor and filter order. This results in high bit growth at the integrator output of the CIC filter and it can be computed as:

$$B_{out} = B_{in} + N * \log_2 RM \quad (4.7)$$

Where,  $B_{out}$  = Output bit width,  $B_{in}$  = Input bit width,  $R$  = Programmable decimation factor,  $M$  = Differential delay (1 or 2), and  $N$  = Order of the CIC filter.

We have designed a CIC filter for four radio communication standards considered in the problem statement. Design parameters for CIC filters are selected as  $N = 3$ ,  $M = 1$  and input bit width as 16-bits. The required programmable decimation factors are 384, 64, 16 and 8 for GSM900, CDMA2000, WCDMA and WiMAX 802.16, respectively. The output bit width of integrators for each radio standard can be computed using Eq.4.7. For GSM900 radio standard, with a maximum decimation value of 384, the output bit width required is 43-bits, and the radio standard WiMAX 802.16 with lowest decimation factor requires output bit width of 25 bits. Since, the designed filter has to support the four radio standards considered, the output bit width is selected as 43-bits.



**Figure 4.3** Structure of the CIC Filter to support Multi-standard Radio Communications

Fig.4.3 shows the structure of CIC filter designed for four radio standards. The input to the CIC filter is resized from 16-bits to 43-bits. This high bit growth at the input of CIC filter limits the frequency of operation of the CIC filter and results in high power dissipation for radio standards with lower decimation factors. These problems are addressed by factorizing the decimation factor into relatively small factors and achieve the required SRC by designing a multi-stage CIC filter.

#### 4.3.2 Multi-stage CIC Filter Design

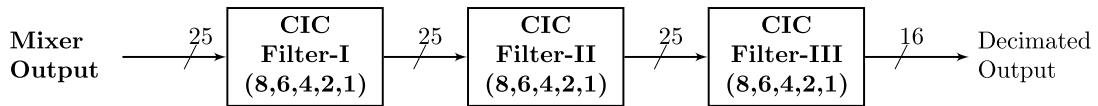
Multi-stage CIC filter design involves factorizing the large integer rate decimation factor by factorizing into relatively small integer factors for reducing the bit growth requirements. For multi-stage CIC filter the required decimation factor  $R$  is expressed as:

$$R = \prod_{i=1}^I R_i \quad (4.8)$$

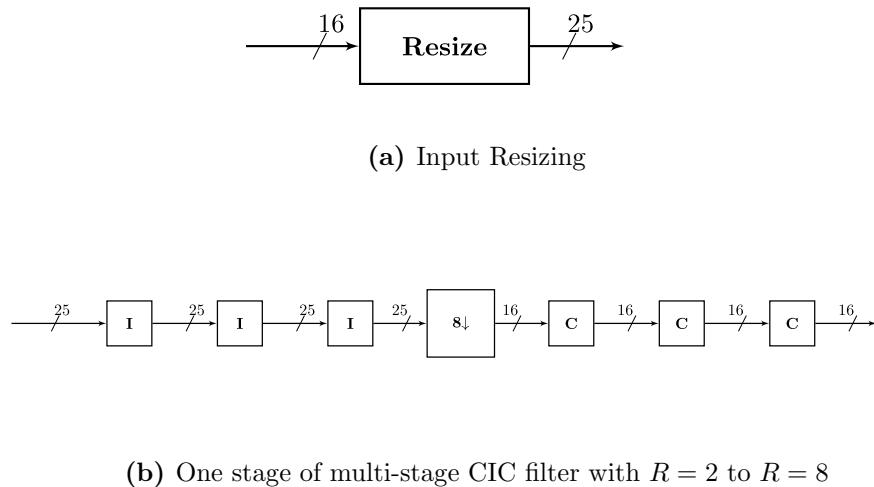
The decimation in the first stage of CIC filter is the smallest decimation factor required, while the largest decimation factor is achieved by processing the signal through all the stages of multi-stage CIC filter.

It may be noted from Table 4.2 (Section 4.2) that the smallest decimation factor required is 8 for WiMAX radio standard and the largest decimation factor required is 384 for GSM900 radio standard. Hence, the largest decimation factor for one stage of CIC

filter is selected as 8 and the number of such CIC filter stages required is three to attain the largest decimation factor of 384. We have considered 16-bits to represent each sample of the input signal and order of the CIC filter in each stage as three with a decimation factor of 8. The bit growth at the input of CIC filter computed using Eq.4.7 is 25-bit which is very low when compared to 43-bit. This results in increased frequency of operation of CIC filter. Fig.4.4 shows the structure of multi-stage CIC filter designed for four radio standards with programmable decimation factors specified in each stage of CIC filter.



**Figure 4.4** Structure of Multi-stage CIC Filter



**Figure 4.5** Structure of the CIC filter with Gain normalized to unity

The area of the CIC filters can be optimized by normalizing the gain of CIC filter to unity as shown in Fig. 4.5. In this structure the number of bits after each integrating stage is reduced by  $\log_2 R$  bits.

**Table 4.3** Performance Comparison of CIC Filter structures on Virtex-6 FPGA

Hardware Resources	Single Stage	Multi-stage
No. of Slice LUTs	321	336
No. of Slice Registers	482	762
No. of LUT-FF Pairs	313	294
Frequency of Operation (in MHz)	453	702

#### 4.3.3 FPGA Implementation

We have designed an integer rate SRC filter as a multi-stage CIC filter with the gain of the filter normalized to unity. This CIC filter is modelled using VHDL and simulated to verify the functionality. These HDL models are synthesized and implemented on Virtex-6 FPGA prototype board using Xilinx ISE 14.7 software.

The device utilization summary and the frequency of operation obtained using Xilinx Synthesis tool for single stage and multi-stage CIC filter on Virtex-6 FPGA is presented in Table 4.3. It is observed that the multi-stage CIC filter can support wideband IF signals when compared with the single stage CIC filter at the expense of slight increment in the hardware resources. Further, the frequency response of the CIC filter is evaluated using FDATOOL in MATLAB.

#### 4.3.4 Frequency Response

The frequency response of the CIC filter is expressed as:

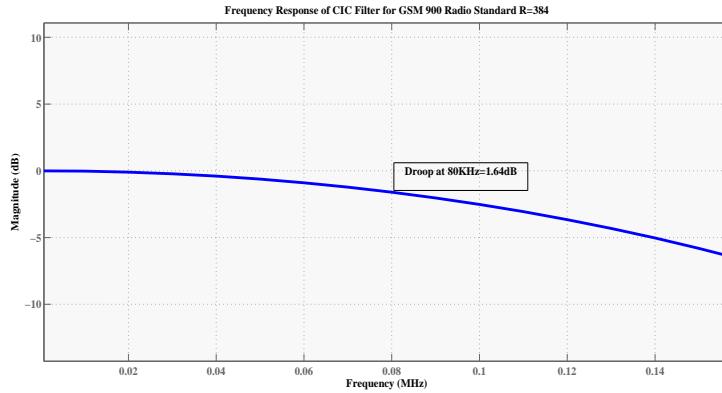
$$H_{CIC}(f) = \left| \frac{\sin(\pi Mf)}{\sin(\frac{\pi f}{R})} \right|^N \quad (4.9)$$

where,  $M$  = Number of differential delay elements,  $R$  = Decimation factor,  $N$  = Order of the CIC filter,  $f = \frac{F}{F_s}$

For large decimation factors as required in SDR receivers and  $M = 1$ , the above equation can be approximated as:

$$H_{CIC}(f) \approx \left| R \frac{\sin(\pi f)}{(\pi f)} \right|^N \quad (4.10)$$

Hence, it may be noted that the frequency response of CIC filter is sinc in nature and introduces gain droop in the pass band of interest. This gain droop has to be restored.



**Figure 4.6** Frequency Response of CIC Filter for GSM900 Radio Standard  
( $R = 384$ , simulated in MATLAB)

**Table 4.4** Computation of CIC Gain Droop for various Radio Standards

Radio Standard	$R$	$F_p$ (in MHz)	$F_{st}$ (in MHz)	$\omega_p$ (in Radians)	$\omega_s$ (in Radians)	Droop (in dB)
WiMAX 802.16	8	4	5	$0.4\pi$	$0.5\pi$	1.74
WCDMA (UMTS)	16	1.2	1.4	$0.472\pi$	$0.56\pi$	2.48
CDMA 2000	64	0.59	0.7	$0.472\pi$	$0.56\pi$	2.48
GSM 900	384	0.08	0.1	$0.384\pi$	$0.48\pi$	1.64

In our design, we have considered IF signal of 80MHz sampled at Nyquist rate of 160MSPs. We have simulated the frequency response of CIC filter with a decimation factor of 384 for GSM900 radio standard and plotted in MATLAB. It may be observed from the frequency response plot shown in Fig.4.6, that the gain droop is 1.64dB at a pass band edge of 80KHz for GSM900 radio standard. Similarly, the gain droop in the pass band of interest is observed for WiMAX 802.16, WCDMA and CDMA2K radio standards and tabulated in Table 4.4 . Finally, this section is concluded by summarizing the advantages and disadvantages of CIC filter.

#### 4.3.5 Summary

In this sub-section, the advantages and disadvantages of CIC filters are summarized. The hardware complexity and computational complexity of the CIC filter is very low and it can be easily reconfigured to attain decimation factor for SDR receivers due to its multiplier-less architecture. However, the bit width of the filter increases with increase in decimation factor and introduces gain droop in the pass band of interest. High bit growth limits the frequency of operation and dissipates high power. Multi-stage CIC filter addresses these issues with a slight increase in hardware complexity. The gain droop in the pass band of interest has to be restored by cascading it with a gain droop compensation filter. Since CIC filters are suitable to attain SRC by large integer rates, it is required to cascade a fractional rate interpolation filter to make sample rate of the signal as the power-of-two multiple of symbol rate. In the subsequent section, we present the design methods for the design of gain droop compensation and interpolation filter.

## 4.4 CIC Compensation and Interpolation filter

As per the specifications considered for the design of multi-standard SDR, the gain droop (Droop) compensation and fractional rate interpolation factor ( $\mu$ ) are presented in Table 4.5. In addition, the spectral characteristics such as pass band edge frequency ( $F_p$  in analog domain,  $\omega_p$  in digital domain), stop band edge frequency ( $F_{st}$  in analog domain,  $\omega_{st}$  in digital domain) are also presented in this table. It may be noted that

**Table 4.5** CIC Gain Droop Compensation and Fractional Rate Interpolation Factors for four Radio Standards [88]

Radio Standard	$F_p$ (in MHz)	$F_{st}$ (in MHz)	$\omega_p$ (in Radians)	$\omega_s$ (in Radians)	Droop (in dB)	$\mu$
WiMAX 802.16	4	5	$0.4\pi$	$0.5\pi$	1.74	0
WCDMA (UMTS)	1.2	1.4	$0.472\pi$	$0.56\pi$	2.48	0.536
CDMA 2000	0.59	0.7	$0.472\pi$	$0.56\pi$	2.48	0.966
GSM 900	0.08	0.1	$0.384\pi$	$0.48\pi$	1.64	0.299

these spectral characteristics vary with the change in the radio communication standard. Hence, reconfigurable architectures are required to design a gain droop compensation filter and fractional rate interpolation filter. In this section, we present different methods employed to restore gain droop and attain symbol rate requirements.

#### 4.4.1 Discrete Gain Compensation and Interpolation filter

##### Discrete Gain Compensation filter

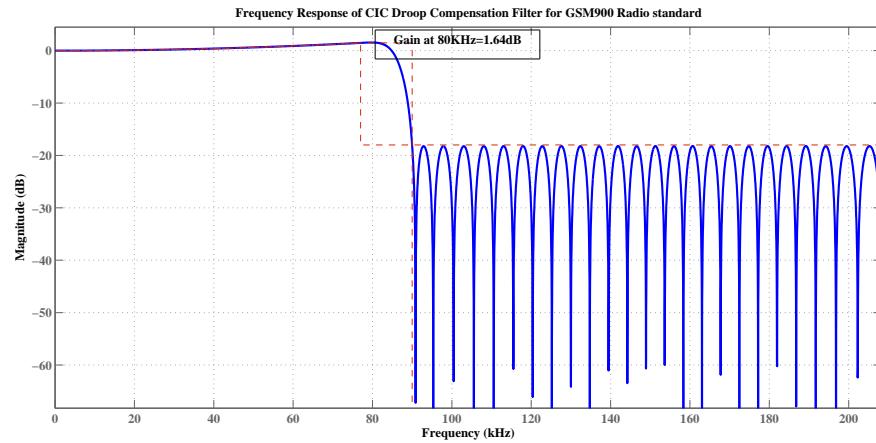
An FIR filter is a simple approach to design a CIC compensation filter. The required frequency response of the FIR filter is expressed as the inverse frequency response of the CIC filter. The frequency response of the CIC decimation filter is approximated as:

$$H_{CIC}(f) \approx \left| R \frac{\sin(\pi f)}{(\pi f)} \right|^N \approx \left| R \frac{\sin(\frac{\omega}{2})}{(\frac{\omega}{2})} \right|^N \quad (4.11)$$

where,  $N$ =Order of CIC filter,  $f$  =Digital frequency,  $R$ =Decimation factor, and  $\omega = 2\pi f$   
Hence, the required frequency response of FIR filter to attain gain droop compensation in the pass band is expressed as:

$$H_{Comp}(\omega) = \begin{cases} \left| \frac{1}{R} \frac{(\frac{\omega}{2})}{\sin(\frac{\omega}{2})} \right|^N, & 0 \leq \omega \leq \omega_p; \\ K_p \left( \frac{\omega - \omega_p}{\omega_s - \omega_p} \right), & \omega_p \leq \omega \leq \omega_s; \\ 0; & \text{elsewhere} \end{cases} \quad (4.12)$$

In our work, the order of the FIR filter is selected as 42 and a gain droop compensation filter for GSM900 radio standard with the required specifications stated in Table 4.5 is designed. We have designed an FIR compensation filter as described in Eq. 4.12 using filter design methods in MATLAB and simulated for its frequency response. The frequency response of the gain compensation filter for GSM900 radio standard is shown in Fig.4.7. It is observed from the frequency response plot that the gain droop of 1.64dB at 80KHz introduced by the CIC filter is restored by cascading a gain droop compensation filter. The compensation filter can be reconfigured to meet the spectral characteristics of another radio standard by redesigning the filter. We have designed the compensation

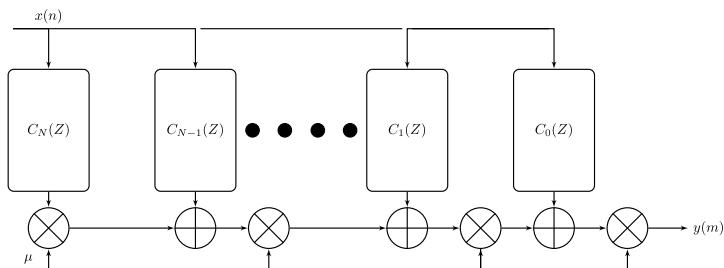


**Figure 4.7** Frequency Response of Gain Compensation Filter designed for GSM900 Radio Standard

filter to restore the gain droop for the radio standards stated in Table 4.5 and attained the spectral characteristics of the desired radio communication standard. In the next sub-section, the design of a fractional rate interpolation filter to meet the symbol rate requirements of the radio standard is presented.

### Fractional Rate Interpolation filter

The symbol rate requirements of the radio standard are met by cascading a fractional rate interpolation filter with the compensated CIC filter. It may be noted from Table 4.5 that the fractional rate SRC factor varies with change in radio communication standard. Hence, it is required to design a reconfigurable SRC filter to support variable fractional rates.



**Figure 4.8** Structure of the Farrow Filter

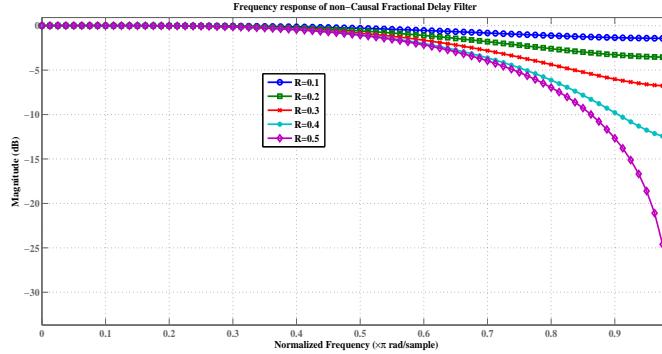
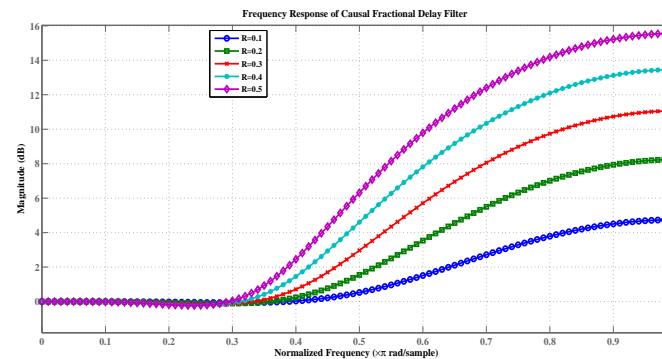
In 1988, C.W. Farrow has proposed a special class of polynomial based filter [42]. An  $L^{th}$  order Farrow filter consists of  $(L + 1)$  sub-filters and computes the output as a function of continuously varying digital delay element,  $\mu$  as shown in Fig.4.8. This filter can be easily reconfigured to attain SRC by variable fractional rates if the sub-filters are designed as fixed coefficient filters. In our work, we employ a numerical interpolation technique based on Lagranges cubic polynomial to design a variable SRC filter for the four radio standards and implemented as a Farrow structure. Eq.4.13 expresses the output of fractional rate SRC filter designed based on Lagranges cubic polynomial interpolation such that it can be realized as a Farrow structure.

$$\begin{aligned}
 x(n + \mu) = & \mu^3 \left( \frac{-1}{6}x(n + 1) + \frac{1}{2}x(n) + \frac{-1}{2}x(n - 1) + \frac{1}{6}x(n - 2) \right) + \\
 & \mu^2 \left( \frac{1}{2}x(n + 1) - x(n) + \frac{1}{2}x(n - 1) + 0 * x(n - 2) \right) + \\
 & \mu^1 \left( \frac{-1}{3}x(n + 1) + \frac{-1}{2}x(n) + x(n - 1) + \frac{-1}{6}x(n - 2) \right) + \\
 & \mu^0 x(n)
 \end{aligned} \tag{4.13}$$

The coefficients of the sub-filters derived using Lagranges cubic interpolation formula are fixed coefficients shown in parentheses of each term of Eq.4.13. It may be noted that the fractional rate interpolation filter designed is a non-causal filter. A causal interpolation filter is derived using Lagranges cubic polynomial interpolation method and the output of the causal Lagranges interpolation filter is expressed as:

$$\begin{aligned}
 x(n + \mu) = & \mu^3 \left( \frac{-1}{6}x(n) + \frac{-1}{2}x(n - 1) + \frac{1}{2}x(n - 2) + \frac{1}{6}x(n - 3) \right) + \\
 & \mu^2 \left( x(n) - \frac{5}{2}x(n - 1) + 2x(n - 2) + \frac{-1}{2}x(n - 3) \right) + \\
 & \mu^1 \left( \frac{11}{6}x(n) - 3x(n - 1) + \frac{3}{2}x(n - 2) + \frac{-1}{3}x(n - 3) \right) + \\
 & \mu^0 x(n)
 \end{aligned} \tag{4.14}$$

The derived non-causal and causal fractional rate Lagranges cubic polynomial interpolation filters with variable fractional delays are simulated in MATLAB to observe their

(a)  $0 \leq \mu \leq 0.5$  (Non-Causal)(b)  $0 \leq \mu \leq 0.5$  (Causal)**Figure 4.9** Frequency response of Lagrange Cubic Polynomial Interpolation Filter

frequency responses. Fig.4.9(a) and Fig.4.9(b) shows the frequency response of the fractional rate interpolation filters with frequency along x-axis and magnitude along y-axis for variable fractional delays,  $\mu$ . It is observed that the gain of fractional rate interpolation filter (magnitude) increases with increase in fractional rate change in the high frequency region leading to high stop band ripple. It is also observed that the gain in the high frequency is very high for the causal filter when compared with the non-causal filter. A joint compensation and interpolation filter designed based on frequency domain compensation and polynomial interpolation is employed to address the above problems.

#### 4.4.2 Proposed Joint Compensation and Interpolation Filter

In this sub-section, we first present the design procedure for joint compensation and interpolation filter based on frequency domain response and polynomial interpolation technique. In our work, we apply this procedure to design a modified joint compensation

and interpolation filter.

This procedure involves designing the joint compensation and interpolation filter [85] as a polynomial interpolation filter using modified Farrow structure [87]. The coefficients of the sub-filters in the Farrow structure are derived from its frequency domain response.

The time domain response of the Farrow filter in terms of impulse response coefficients of the FIR filter is

$$h(n + \mu) = \sum_{l=0}^L c_l(n)(2\mu - 1)^l \quad (4.15)$$

and the corresponding frequency response is

$$\begin{aligned} H(j\omega) &= \frac{1}{F_s} \sum_{l=0}^L C_l(j\omega) \int_0^1 (2\mu - 1)^l e^{(-j\omega\mu)} d\mu \\ &= \frac{1}{F_s} \sum_{l=0}^L C_l(j\omega) W_l(j\omega) \end{aligned} \quad (4.16)$$

where,  $F_s$  = Input sampling frequency of a specific radio standard,

$C_l(\omega)$  = Frequency response of sub-filters in Farrow structure,

$W_l(\omega)$  = Frequency response of windowing function,  $(2\mu - 1)^l$ ,

$\mu$  = Required Fractional delay

The frequency response of the sub-filters in the Farrow structure and windowing function in Eq.4.16 are expressed as:

$$C_l(j\omega) = \begin{cases} 2 \sum_{n=0}^{\frac{N}{2}-1} c_l(n) \cos \left( \left( \frac{N-1}{2} - n \right) \omega \right), & l = \text{even}; \\ 2 \sum_{n=0}^{\frac{N}{2}-1} c_l(n) \sin \left( \left( \frac{N-1}{2} - n \right) \omega \right), & l = \text{odd} \end{cases} \quad (4.17)$$

$$W_l(j\omega) = \begin{cases} 2 \sum_{k=0}^l k! \binom{l}{k} \left( \frac{\sin \left( \frac{\omega}{2} + \frac{k\pi}{2} \right)}{\left( \frac{\omega}{2} \right)^{k+1}} \right), & l = \text{even}; \\ 2 \sum_{k=0}^l k! \binom{l}{k} \left( \frac{\cos \left( \frac{\omega}{2} + \frac{k\pi}{2} \right)}{\left( -\frac{\omega}{2} \right)^{k+1}} \right), & l = \text{odd}; \end{cases} \quad (4.18)$$

From Eq.4.17 and Eq.4.18, it is observed that the only unknowns in these equations are Farrow coefficients,  $c_l(n)$ . The frequency response of the Farrow structure is equated to the desired response and the unknown Farrow coefficients are evaluated using linear

programming such that the error between the two is minimized either in the least squares sense or minimax sense. In our work, the desired response is CIC compensation response.

The frequency response of CIC compensation filter is approximated with a second order frequency domain polynomial in the pass band, linear function in the transition band and zero in the stop band [85]. Mathematically, the approximated frequency response is

$$H_{Comp}(\omega) = \begin{cases} 1 + A\omega^2 + B\omega, & 0 \leq \omega \leq \omega_p; \\ K_p \left( \frac{\omega - \omega_p}{\omega_s - \omega_p} \right), & \omega_p \leq \omega \leq \omega_s; \\ 0; & \text{elsewhere} \end{cases} \quad (4.19)$$

where,  $K_p$  is the required peak pass band gain.

It is observed that this approximation requires evaluation of two polynomial coefficients  $A$  and  $B$ , which changes with the change in the spectral characteristics. To address this problem we propose a modified joint compensation and interpolation filter. The desired frequency response of CIC compensation filter for the proposed filter is expressed as:

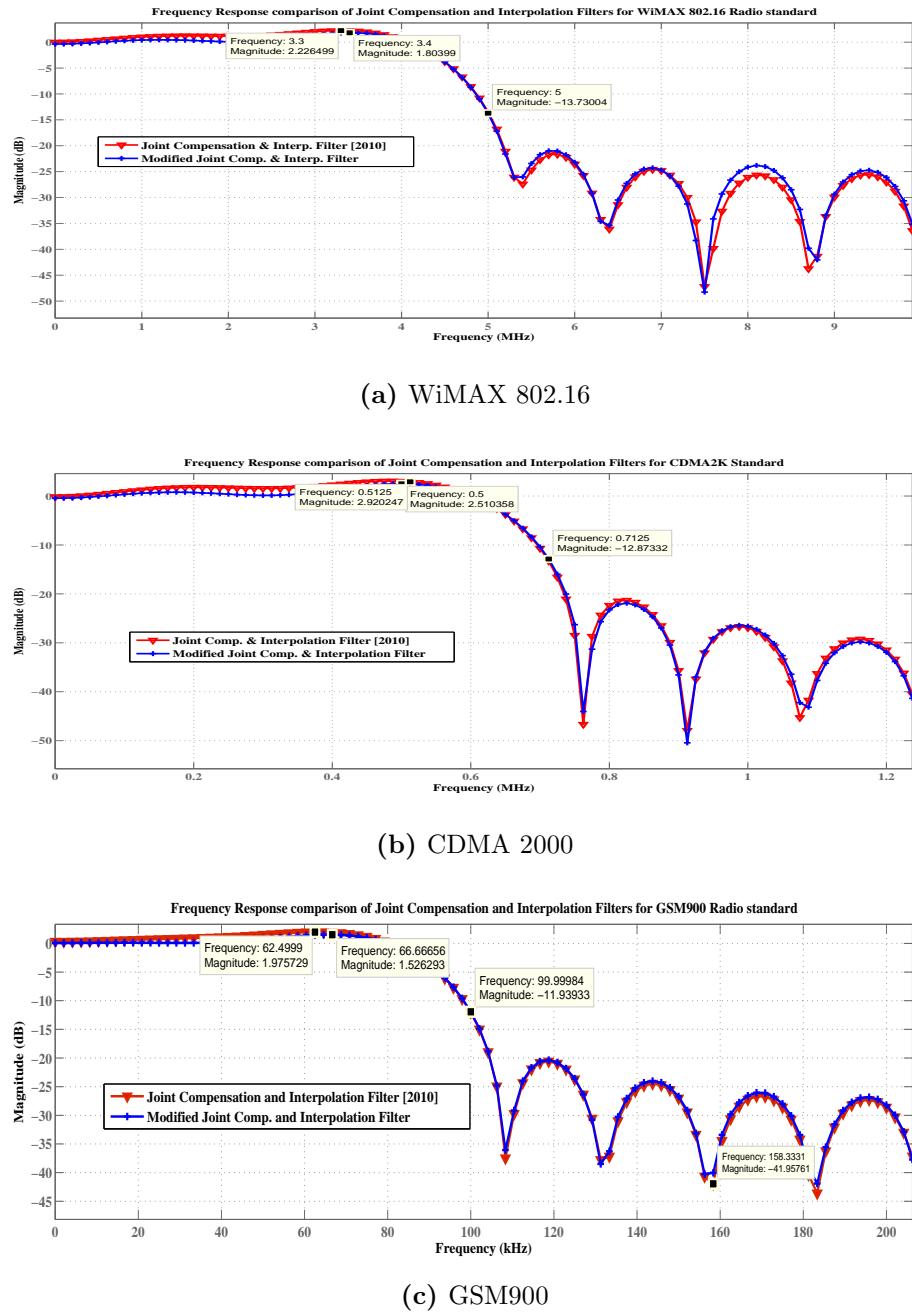
$$H_{Comp}(\omega) = \begin{cases} \left( \frac{\omega}{\sin \frac{\omega}{2}} \right)_{CIC}^N, & 0 \leq \omega \leq \omega_p; \\ K_p \left( \frac{\omega - \omega_p}{\omega_s - \omega_p} \right), & \omega_p \leq \omega \leq \omega_s; \\ 0; & \text{elsewhere} \end{cases} \quad (4.20)$$

In Eq.4.20 the frequency response of CIC compensation filter is expressed as the normalized inverse frequency response of CIC filter in the pass band region in contrast to the second order polynomial approximation in pass band region of the joint compensation and interpolation filter expressed in Eq.4.19.

In our work, we have considered the order of CIC filter,  $N_{CIC} = 3$ , order of Farrow filter,  $L = 3$  with  $N = 32$  coefficients in each sub-filter of the Farrow structure. The error between the frequency response of the Farrow structure and the desired frequency response is computed as:

$$E(\omega) = H(\omega) - H_{Comp}(\omega) \quad (4.21)$$

The error,  $E(\omega)$  in Eq.4.21 is minimized in the least squares sense and the unknown coefficients of the Farrow structure are evaluated using linear programming technique in MATLAB. Then the frequency response of the modified joint compensation and interpolation filter is simulated using filter design and analysis tool in MATLAB.



**Figure 4.10** Frequency Response of Proposed Modified Joint compensation and Interpolation Filter and Joint Compensation and Interpolation Filter

## Frequency Response

The modified joint compensation and interpolation filter is designed to restore the gain droop of WiMAX, CDMA2000 and GSM900 radio standards as specified in Table 4.5. The frequency response of the proposed modified joint compensation and interpolation filter and existing joint compensation and interpolation filter is obtained using FDATOOL in MATLAB for each of the radio communication standards considered in this work and the same is plotted in Fig.4.10. From these plots, the value of peak gain frequency measured for three radio communication standards are presented in Table 4.6. It may be observed that the frequency at which the peak gain achieved by the proposed and existing [85] joint compensation and interpolation filters for each of the radio communication standards considered is less than that required as per the specifications (See Table 4.5). For example, second column of Table 4.6 shows 3.3MHz and 3.4MHz are the peak gain frequency values obtained for WiMAX radio standard using the existing compensation filter [85] and the proposed filter, respectively. These values are less than the expected value of 4MHz as per the specifications. Hence, it can be concluded that these filters results in frequency warping.

**Table 4.6** Comparison of Peak Gain Frequency (in MHz) of Existing and Proposed Joint Compensation and Interpolation Filters

Frequency at Peak Gain	WiMAX 802.16	CDMA-2000	GSM-900
Required	4	0.59	0.08
Joint Comp. [2010] [85]	3.3	0.5125	0.06249
Proposed Joint Comp.	3.4	0.5	0.06666

In addition, the value of peak gain is measured from the frequency response plotted in Fig.4.10 for three radio communication standards and the same are presented in Table 4.7. These values are used to compute the peak gain error and also presented in the last column of the Table 4.7. It may be observed that the peak gain of the proposed and the existing [85] filters is more than the required value as per the specifications (Table 4.5). This results in ripples in the pass band. For example, the peak gain expected for WiMAX radio standard is 1.74dB. However, the peak gain of existing [85] and the

**Table 4.7** Comparison of Peak Gain Error (in dB) of Existing and Proposed Joint Compensation and Interpolation Filters

Radio Standard	Peak Gain			Peak Gain Error	
	Required	Joint Comp. [85]	Proposed	Joint Comp. [85]	Proposed
WiMAX	1.74	2.22	1.80	0.48	0.06
CDMA2000	2.48	2.92	2.51	0.44	0.03
GSM900	1.64	1.97	1.53	0.33	0.11

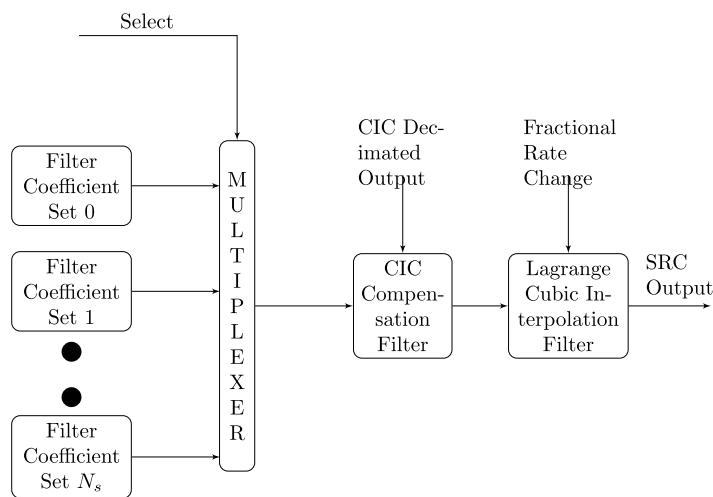
proposed filters are 2.22dB and 1.8dB, resulting in gain error of 0.48dB and 0.06dB, respectively. Hence, it can be concluded that the proposed filter results in low pass band ripple compared to the filter considered for comparison [85]. It is also observed that the stop band attenuation characteristics of the proposed filter and the filter considered for comparison are almost equal for three radio communication standards (See Fig.4.10). However, the obtained stop band attenuation at the required stop band edge frequency of the various radio standards is slightly low for the joint compensation and interpolation filters. Hence, it can be concluded that the filter designed employing proposed method attains better spectral characteristics when compared with the existing method in the literature.

#### 4.4.3 Architectures

##### Discrete Compensation and Interpolation Filter

The architecture of the Discrete Compensation and Interpolation Filter to support multi-standard radio communications is shown in Fig.4.11. It consists of  $N_s$  look up tables to store filter coefficients pertaining to  $N_s$  radio standards to attain their spectral characteristics, multiplexer to select one of the filter coefficient sets based on the radio standard into consideration, a discrete CIC compensation filter to restore the gain droop in the pass band of interest and an interpolation filter to make the sample rate of the signal as power-of-two multiple of symbol rate. The discrete CIC compensation filter is realized using symmetric FIR structure and a fractional rate interpolation filter is realized

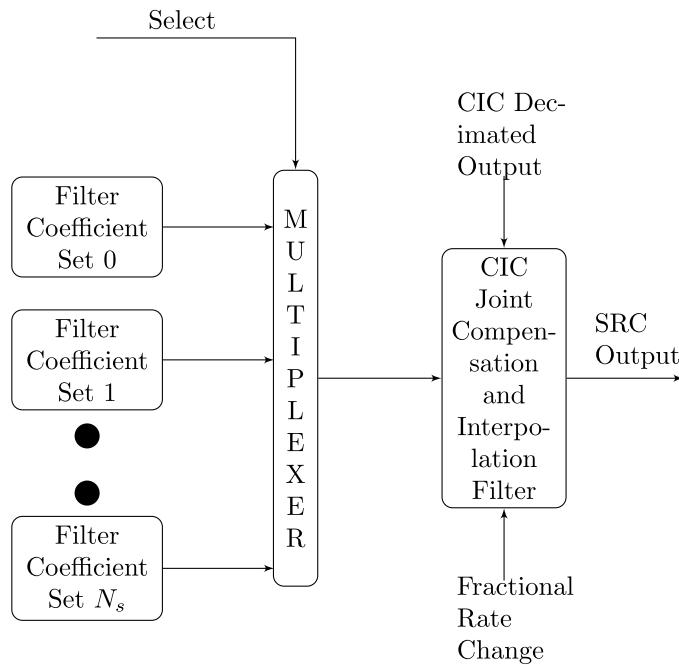
using Farrow structure. The filter coefficients to design these filters are computed using the filter design methods presented in Section 4.4.1. The hardware complexity of these filters is computed in terms of number of MAC units. It may be noted that the hardware complexity of a  $N^{th}$  order symmetric FIR filter structure is approximately equal to  $\frac{N}{2}$  and that of  $L^{th}$  order polynomial interpolation filter realized using Farrow structure is  $(L+1)(L+1)$  MAC units. In our work, the order of CIC compensation filter is selected as,  $N = 42$  (43-filter coefficients) and the order of polynomial for fractional rate interpolation filter is selected as,  $L = 3$ , resulting in a hardware complexity of 38 MAC units.



**Figure 4.11** Discrete Compensation and Interpolation Filter Architecture for Multi-standard Radio Receiver

### Proposed Joint Compensation and Interpolation Filter

The architecture of the proposed Joint Compensation and Interpolation filter to support multi-standard radio communications is shown in Fig.4.12. It consists of  $N_s$  look up tables to store filter coefficients pertaining to  $N_s$  radio standards to attain their spectral characteristics, multiplexer to select one of the filter coefficient sets based on the radio standard into consideration, a CIC Joint compensation and interpolation filter to restore the gain droop in the pass band of interest and attain symbol rate requirements of radio communication standards. The filter coefficients are computed using either of the joint compensation and interpolation filter design methods presented in Section 4.4.2.



**Figure 4.12** Proposed Joint Compensation and Interpolation Filter Architecture for Multi-standard Radio Receiver

The hardware complexity of this filter is computed as follows. Since the realization of this filter involves an  $L^{th}$  order Farrow structure with  $(L+1)$  sub-filters and  $N$  filter coefficients in each sub-filter, the hardware complexity is computed as  $(L + 1)(N) + (L + 1)$  MAC units. Due to symmetric filter coefficients the hardware complexity is reduced by a factor of two. In our work, the order of Farrow structure is selected as  $L = 3$  and the number of filter coefficients as  $N = 32$ , resulting in a hardware complexity of 68 MAC units.

## 4.5 Implementation Results

The filters required to perform sample rate conversion in SDR receivers are modelled using VHDL. These RTL models are simulated and synthesized using Xilinx Vivado Design Suite to verify the functionality and the RTL schematic. In addition, the synthesized netlist is also implemented on Kintex-7 Family XC7K325t-2ffg900 FPGA. The designed SRC filters are functionally verified for WiMAX, CDMA2000 and GSM900 radio standards and the device utilization summary obtained by implementing these filters are

**Table 4.8** Hardware Complexity Comparison of CIC Compensation and Interpolation Filters implemented on Kintex-7 XC7K325t-2ffg900 FPGA

Hardware Resources	Discrete Comp.	Proposed Joint Comp.
No. of Slice LUTs	23804	50646
No. of Slice Registers	24452	28371
No. of LUT-FF Pairs	17969	21531

presented in Table 5.4. It is observed that the implementation of SRC filter based on discrete compensation and interpolation filter requires less hardware resources when compared to the SRC filter based on proposed joint compensation and interpolation filter. However, the spectral characteristics discrete compensation and interpolation filter are poor when compared to the spectral characteristics of the proposed joint compensation and interpolation filter as presented in Table 4.5. In addition, it is also observed that both the filters require offline computations to support any existing or forthcoming radio standard.

## 4.6 Conclusions

In this chapter, SRC filter based on low complexity CIC filters to support multi-standard radio communications is realized. The integer rate decimation filter is designed as a single stage CIC filter and multi-stage CIC filter with order of the filter being three and decimation factors varying from 2 to 512 and implemented on Kintex-7 FPGA. From the FPGA implementation results, it is observed that the multi-stage CIC filters can support wideband IF signals when compared with the single stage CIC filter. The frequency response of the designed multi-stage CIC filter is simulated in MATLAB. It is observed that CIC filters introduce gain droop in the pass band of interest. Discrete compensation filter based on the inverse frequency response of the CIC filter and fractional rate interpolation filter based on Lagrange's polynomial interpolation is designed and simulated in MATLAB to observe their frequency response. It is observed that the frequency response of the Lagrange interpolation filter produces high gain in the high frequency region. A

modified joint compensation and interpolation filter based on the design of existing joint compensation and interpolation filter is proposed. These filters are simulated in MATLAB to compute their frequency response. From the simulation results, it is observed that the spectral characteristics of the proposed joint compensation and interpolation filter are improved when compared with the existing compensation and interpolation filters. However, FPGA implementation results show it requires more hardware resources when compared to the discrete compensation and interpolation filter. It may also be noted that the existing CIC compensation and interpolation filters and the proposed filter needs to be redesigned with the change in radio communications standard, which involves high reconfiguration overhead. Design of reconfigurable SRC filter with tuneable spectral characteristics and minimum reconfiguration overhead is presented in the next chapter.

# Chapter 5

## SVD based Reconfigurable SRC Filter for SDR Receivers

This chapter presents design of reconfigurable SRC filter with minimum reconfiguration overhead to support multi-standard radio communications as required for the design of SDR receivers. In this work, a tuneable joint compensation and interpolation filter based on SVD-VDF is proposed to support multi-standard radio communications with minimum reconfiguration overhead. The proposed SVD-VDF based joint compensation and interpolation filter is simulated in MATLAB to compare its frequency response with the existing compensation and interpolation filters available in the literature. Further, the hardware complexity of the proposed SVD-VDF filter is computed analytically and compared with the existing architectures. In addition, VHDL model for SVD-VDF based reconfigurable SRC filter is developed to verify its functionality and implemented on Kintex-7 FPGA to compare the hardware complexity with the existing SRC filters.

### 5.1 Introduction

It may be observed that coefficient-less CIC filters are suitable architectures to attain sample rate change by large integer factors and easily reconfigurable. However, these filters introduce gain droop in the pass band of interest and cannot achieve SRC by fractional rate. The gain droop compensation and interpolation filters presented in Chapter 4 require offline computations to support a new radio communication standard

resulting in high reconfiguration overhead. Hence, it is required to design a variable digital filter with reconfigurable spectral characteristics and minimum reconfiguration overhead to support multi-standard radio communications.

A straight forward method to design variable digital low pass filters with variable fractional delay (VFD) based on singular value decomposition (SVD) of the desired variable frequency response is presented in [59]. SVD technique generates complex vectors and real vectors. The complex vectors are considered as the frequency responses of the 1-D constant FIR filters designed with symmetrical or anti-symmetrical coefficients to reduce computational complexity. Similarly, real vectors can be regarded as the desired values of 1-D polynomials with either even symmetry or odd symmetry. The SVD-VDF based VFD filter with one variable parameter is extended to design a VDF with multiple variable spectral parameters. In this method, the original 1D-variable filter is decomposed using SVD algorithm and implemented using constant coefficients 1D-FIR filters and multi-dimensional polynomial approximations obtained by constant vector-array decomposition (VAD) [60], [61].

## 5.2 Singular Value Decomposition (SVD) Algorithm

The singular value decomposition of a  $m \times n$  matrix  $A$  is of the form

$$\begin{aligned} SVD(A) &= U\Sigma V^T \\ &= \sum_{i=1}^r \sigma_i u_i v_i \end{aligned} \tag{5.1}$$

where,  $u_i$  = Eigen vector of  $AA^T$ ,

$v_i$  = Eigen vector of  $A^T A$ ,

$\sigma_i$  = Eigen values of  $AA^T$  or  $A^T A$ ,

$r$  = Rank of matrix,  $A$

The orthogonal matrices  $U, V$  span the row space and column space of matrix  $A$ , respectively. The square root of non-zero Eigen values are the singular values of matrix  $A$  and the elements in the matrix  $\Sigma$  satisfies the relation,

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{r-1} \geq \sigma_r \geq 0 \tag{5.2}$$

Neglecting the low magnitude singular values, the matrix  $A$  with rank  $r$  can be approximated with a relatively lower rank matrix as

$$A \approx \sum_{i=1}^k u_i \sigma_i v_i, \quad k < r \quad (5.3)$$

However, it is difficult to choose an appropriate value of  $k$  that approximates the matrix  $A$ . Consider all the singular values such that a large variation is observed between two consecutive singular values and neglect all the singular values with low variation between them. Stabilization of singular values represents much of the data is contaminated with noise and hence, not useful.

Singular value decomposition algorithm is applied in the area of image processing, signal processing etc., for noise reduction. In digital signal processing, if a matrix  $A$  represents a noisy signal then compute the SVD of the matrix  $A$  and discard all the smaller and stable singular values of  $A$ . Hence, matrix  $A$  with low rank  $k$  represents a filtered signal with less noise and expressed as:

$$A_k = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \dots + \sigma_k u_k v_k^T \quad (5.4)$$

It is observed that the matrix  $U$  spans the row space and the matrix  $V$  spans the column space of matrix  $A$ . This is the basis to design a variable digital filter (VDF). A VDF is one in which the spectral parameters of the filter such as pass band edge frequency, stop band edge frequency, magnitude response, fractional delay etc. are variable. In the design of VDF, if the rows of matrix  $A$  represent the frequency response of the filter over the whole digital frequency range and the columns represent the variation of frequency response due to variation in its spectral parameters, then the singular vectors of matrix  $U$  represents the frequency response and the singular vectors of matrix  $V$  represents the effect of spectral parameters on the frequency response.

### 5.3 SVD-VDF Design

The method for designing VDFs with tuneable magnitude response and variable fractional delay is explained as:

Let  $\omega$  be the digital frequency spectrum of a signal varying between  $\pi$  and  $-\pi$ , and  $K$  be the number of spectral parameters which controls the spectral characteristics of the VDF. The magnitude response of the VDF with variable fractional delay in terms of  $m$  spectral parameters is expressed as:

$$H(\omega) = M(\omega, \psi_1, \psi_2, \dots, \psi_{K-1}) e^{-j\omega\psi_K} \quad (5.5)$$

where,  $M(\omega, \psi_1, \psi_2, \dots, \psi_{K-1})$  is the magnitude response of the required filter and  $\psi_K$  is the spectral parameter to control the fractional delay.

The digital frequency range and the range of  $K$  spectral parameters in the design of VDF are selected as:

$$\begin{aligned} \omega \in [-\pi, \pi]; \psi_1 &= [-\psi_{1min}, \psi_{1max}]; \psi_2 = [-\psi_{2min}, \psi_{2max}]; \psi_3 = [-\psi_{3min}, \psi_{3max}]; \\ &\dots \dots \psi_K = [-\psi_{Kmin}, \psi_{Kmax}] \end{aligned}$$

### Design steps:

#### Step 1:

Discretise the spectral parameters uniformly at large instants,  $L, M_1, M_2, \dots, M_K$  as shown below.

$$\begin{aligned} \omega(l) &= -\pi + 2\pi \frac{(l-1)}{(L-1)}, & l &= 0, 1, 2, \dots, L-1 \\ \psi_1(m_1) &= \psi_{1min} + \frac{m_1-1}{M_1-1}(\psi_{1max} - \psi_{1min}), & m_1 &= 0, 1, 2, \dots, M_1-1 \\ \psi_2(m_2) &= \psi_{2min} + \frac{m_2-1}{M_2-1}(\psi_{2max} - \psi_{2min}), & m_2 &= 0, 1, 2, \dots, M_2-1 \\ &\dots \\ \psi_K(m_k) &= \psi_{Kmin} + \frac{m_k-1}{M_K-1}(\psi_{Kmax} - \psi_{Kmin}), & m_k &= 0, 1, 2, \dots, M_K-1 \end{aligned} \quad (5.6)$$

#### Step 2:

Construct a multi-dimensional complex array  $\mathbf{H}$  according to the specifications of 1-D frequency response as expressed by Eq.5.5.

**Step 3:**

Transform the multi-dimensional complex array  $\mathbf{H}$  into 2-D complex array  $A$  expressed as

$$A = H_{comp}(\omega(l), \psi(m)), l \Rightarrow l, m \Rightarrow (m_1, m_2, \dots, m_K) \quad (5.7)$$

$$\text{where, } m = (m_1 - 1) \prod_{k=2}^K M_k + (m_2 - 1) \prod_{k=3}^K M_k + \dots + (m_{k-1} - 1) M_k + m_k.$$

**Step 4:**

Apply SVD on 2-D matrix  $A$  such that it can be expressed as sum of products of weighted singular matrices  $U$  and  $V$ . Hence, the matrix  $A$  is expressed as:

$$SVD(A) = U\Sigma V^T = \sum_{i=1}^r \sigma_i u_i v_i^* = \sum_{i=1}^r \sqrt{\sigma_i} u_i * \sqrt{\sigma_i} v_i^* = \sum_{i=1}^r \tilde{u}_i \tilde{v}_i^* \quad (5.8)$$

where,  $\sqrt{\sigma_i}$  is the weight of each singular vector  $u_i$  and  $v_i$  and it follows the relation stated by Eq.5.4.

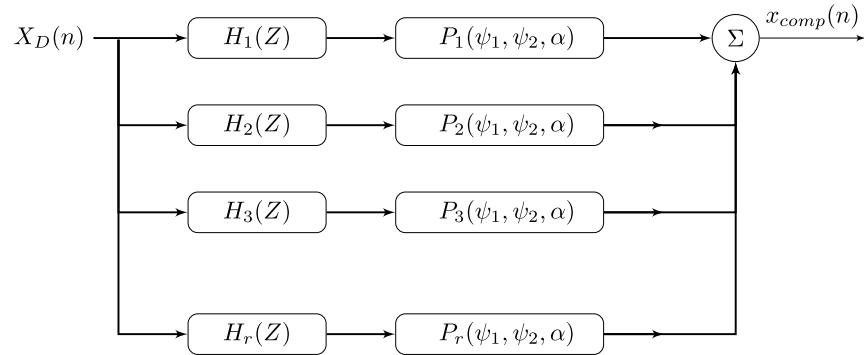
Neglecting the low weight singular vectors, the matrix  $A$  is approximated as

$$A \approx \sum_{i=1}^k \sigma_i u_i v_i, \quad k < r \quad (5.9)$$

The value of  $k$  is selected as described in section 5.1.

**Step 5:**

Since  $U$  spans the row space of  $A$ , the singular vectors in  $U$  represents the effect of frequency variation on magnitude of  $A$ . Hence, the singular vectors in  $U$  are realized as 1-D FIR filters with  $\sqrt{\sigma_i} u_i$  being their magnitude responses. As  $V$  spans the column space of  $A$ , the singular vectors in  $V$  represents the effect of spectral parameters in variation of magnitude of  $A$ . Hence, the singular vectors in  $V$  are realized as a  $K$ -dimensional polynomial in terms of spectral parameters.



**Figure 5.1** Architecture of SVD based Variable Digital Filter

**Step 6:**

Connect these 1-D constant filters with their respective  $K$ -D polynomials to form a variable digital filter. Fig. 5.1 shows the architecture of a SVD- based VDF with  $k$  constant sub-filters connected with  $k$   $K$ -D polynomials.

It is observed that, in the architecture of SVD-VDF the spectral characteristics of VDF can be easily reconfigured by varying the spectral parameters. Hence, it can be concluded that the SVD-VDF involves minimum reconfiguration overhead for reconfiguring the spectral characteristics of the filter. In our work, we employ this SVD based VDF architecture to design a reconfigurable joint compensation and interpolation filter.

## 5.4 Proposed SVD-VDF based Joint Compensation and Interpolation Filter

In this section, we present the design of proposed joint compensation and interpolation filter based on SVD based VDF.

## Specifications

The magnitude response of CIC filter is expressed as:

$$H(\omega) = \left| \frac{\sin(\frac{\omega}{2})}{\sin(\frac{\omega}{2R})} \right|^N \approx \left| R \frac{\sin(\frac{\omega}{2})}{(\frac{\omega}{2})} \right|^N \quad (5.10)$$

where,  $F$  = Analog Frequency (in Hertz),

$F_s$  = Input sampling frequency,

$R$  = Integer rate decimation Factor of CIC filter,

$N$  = Order of the CIC filter

$\omega$  = Digital frequency (in Radians)

The gain normalized magnitude response of CIC filter is indicated as:

$$H(\omega) \approx \left| \frac{\sin(\frac{\omega}{2})}{(\frac{\omega}{2})} \right|^N \quad (5.11)$$

We have considered an IF signal of 80MHz with an input sampling frequency of 160MSPs and four radio standards WiMAX 802.16, WCDMA, CDMA2000 and GSM900. The gain droop observed at the pass band edge frequency for four radio standards with their respective integer rate decimation factors is computed using Eq.5.11. The specifications for the required gain droop compensation and the fractional rate interpolation factor (See Table 4.5) for the respective radio standards are tabulated in Table 5.1.

**Table 5.1** Specifications for Gain Droop Compensation and Fractional Rate Interpolation Factors for various Radio Standards

Radio Standard	$F_p$ (in MHz)	$F_{st}$ (in MHz)	$\omega_p$ (in Radians)	$\omega_s$ (in Radians)	Droop (in dB)	$\mu$
WiMAX 802.16	4	5	$0.4\pi$	$0.5\pi$	1.74	0
WCDMA (UMTS)	1.2	1.4	$0.472\pi$	$0.56\pi$	2.48	0.536
CDMA 2000	0.59	0.7	$0.472\pi$	$0.56\pi$	2.48	0.966
GSM 900	0.08	0.1	$0.384\pi$	$0.48\pi$	1.64	0.299

For the specifications stated in Table 5.1, we derive the parameters required to design a SVD-VDF based joint compensation and interpolation filter.

### Design of SVD-VDF based Joint Compensation and Interpolation Filter

The frequency response of the filter to achieve required gain droop compensation is expressed as:

$$H_{Comp}(\omega) = \begin{cases} \left( \frac{\frac{\omega}{2}}{\sin \frac{\omega}{2}} \right)_{CIC}^N, & 0 \leq \omega \leq \omega_p; \\ K_p \left( \frac{\omega - \omega_p}{\omega_s - \omega_p} \right), & \omega_p \leq \omega \leq \omega_s; \\ 0; & \text{elsewhere} \end{cases} \quad (5.12)$$

where,  $\omega_p$  = Pass band edge frequency,  $\omega_s$  = Stop band edge frequency and  $K_p$  = Peak pass band gain at pass band edge frequency.

It is observed from Table 5.1 that for multi-standard radio communications, spectral parameters such as pass edge frequency, stop band edge frequency and fractional rate interpolation factor vary with change in radio communication standard. Hence, we employ SVD-VDF technique and express the desired frequency response of the filter in terms of variable spectral parameters as:

$$H(\omega) = M(\omega, \psi_1, \psi_2) e^{-j\omega\alpha} \quad (5.13)$$

and for the design of compensation filter,

$$M(\omega, \psi_1, \psi_2) = H_{comp}(\omega) \quad (5.14)$$

The pass band edge frequency, stop band edge frequency and fractional delay are expressed as variable spectral parameters given by

$$\begin{aligned} \omega_p &= \omega_1 + \psi_1 \\ \omega_s &= \omega_2 + \psi_2 \\ \alpha &= \mu - \frac{1}{2} \end{aligned}$$

where,  $\omega_1$  = Minimum fixed pass band edge frequency;

$\psi_1$  = Spectral parameter to vary pass band edge frequency;

$\omega_2$  = Minimum fixed stop band edge frequency;

$\psi_2$  = Spectral parameter to vary stop band edge frequency and

$\alpha$  = Fractional delay parameter

These spectral parameters are derived for the four radio standards presented in Table 5.1. It is observed from Table 5.1 that, the required minimum pass band edge frequency is  $0.384\pi$  for GSM900 radio standard and the maximum pass band edge frequency is  $0.472\pi$  for WCDMA/CDMA 2000 radio standards. The minimum fixed pass band edge frequency is selected as  $0.3\pi$  and the spectral parameter leading to change in the pass band edge frequency varies between  $0.084\pi$  to  $0.172\pi$ . Similarly, the required minimum stop band edge frequency is  $0.48\pi$  for GSM900 radio standard and the maximum stop band edge frequency is  $0.56\pi$  for WCDMA/CDMA 2000 radio standards. The minimum fixed stop band edge frequency is selected as  $0.4\pi$  and hence, the spectral parameter to change the stop band edge frequency varies between  $0.08\pi$  to  $0.16\pi$ . From Table it is observed that the fractional delay parameter may be mapped between 0.036 and 0.466. Hence, the range of parameters selected to design SVD-VDF are summarized as:

Digital frequency range,  $\omega = [-\pi, \pi]$ ;

Minimum fixed pass band edge frequency,  $\omega_1 = 0.3\pi$ ;

Minimum fixed stop band edge frequency,  $\omega_2 = 0.4\pi$ ;

Range of spectral parameter to vary pass band edge frequency,  $\psi_1 = [-0.2\pi, 0.2\pi]$ ;

Range of spectral parameter to vary stop band edge frequency,  $\psi_2 = [-0.2\pi, 0.2\pi]$ ;

Range of Fractional delay parameter,  $\alpha = [-0.5, 0.5]$

## Design steps

### Step 1:

The digital frequency, spectral parameter to vary pass band edge frequency, spectral parameter to vary stop band edge frequency and the fractional delay parameter are uniformly sampled at  $L$ ,  $M_1$ ,  $M_2$  and  $M_3$  points, respectively. In our work, we have selected  $[L, M_1, M_2, M_3] = [600, 11, 11, 10]$  points and discretised the spectral parameters as

$$\omega(l) = -\pi + 2\pi \frac{(l-1)}{(599)}, \quad l = 0, 1, 2, \dots, 599$$

$$\psi_1(m_1) = -0.2\pi + \frac{m_1-1}{10}(0.4\pi), \quad m_1 = 0, 1, 2, \dots, 10$$

$$\psi_2(m_2) = -0.2\pi + \frac{m_2-1}{10}(0.4\pi), \quad m_2 = 0, 1, 2, \dots, 10$$

$$\psi_3(m_3) = -0.5 + \frac{m_3 - 1}{9}, \quad m_k = 0, 1, 2, \dots, 9 \quad (5.15)$$

**Step 2:**

A four dimensional complex array  $\mathbf{H}$  of size  $600 \times 11 \times 11 \times 10$  according to the specifications of 1-D frequency response of the joint compensation and interpolation filter,  $H_{comp}(\omega)$  is formed.

$$\mathbf{H} = H_{comp}(\omega, \psi_1, \psi_2) e^{(-j\omega\alpha)} \quad (5.16)$$

**Step 3:**

The obtained four-dimensional complex array  $[\mathbf{H}]$  is transformed into 2-D array  $\mathbf{A}$  of size  $600 \times 1210$  as stated by Eq.5.17.

$$A = H_{comp}(\omega(l), \psi(m)), l \Rightarrow l, m \Rightarrow (m_1, m_2, m_3) \quad (5.17)$$

$$\text{where, } m = (m_1 - 1) \prod_{k=2}^3 M_k + (m_2 - 1) M_k + m_k$$

**Step 4:**

The 2-D matrix  $A$  is decomposed using SVD algorithm into singular matrices  $U$  and  $V$  with sizes  $600 \times 600$  and  $1210 \times 1210$ , respectively and a diagonal matrix,  $\Sigma$  with singular values of size  $600 \times 1210$  is formed. Hence, the matrix  $A$  is expressed as:

$$SVD(A) = U\Sigma V^T = \sum_{i=1}^r \sigma_i u_i v_i^* = \sum_{i=1}^r \sqrt{\sigma_i} u_i * \sqrt{\sigma_i} v_i^* = \sum_{i=1}^r \tilde{u}_i \tilde{v}_i^* \quad (5.18)$$

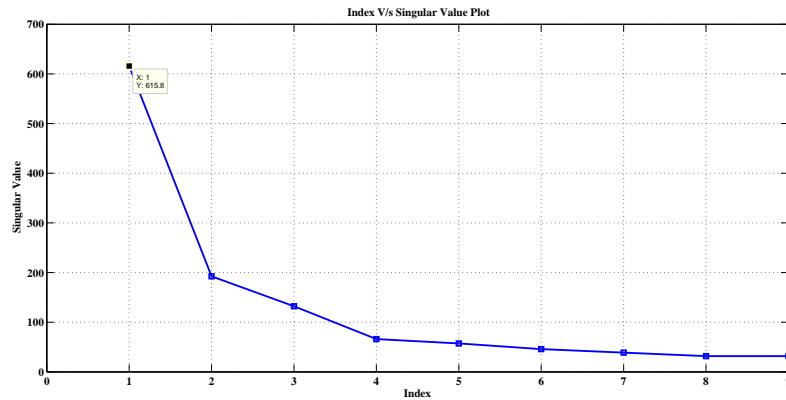
where,  $[U]_{600 \times 600}$  = Magnitude response of constant 1-D sub-filters

$[V]_{1210 \times 1210}$  = Fit a 3-D polynomial in terms of spectral parameters

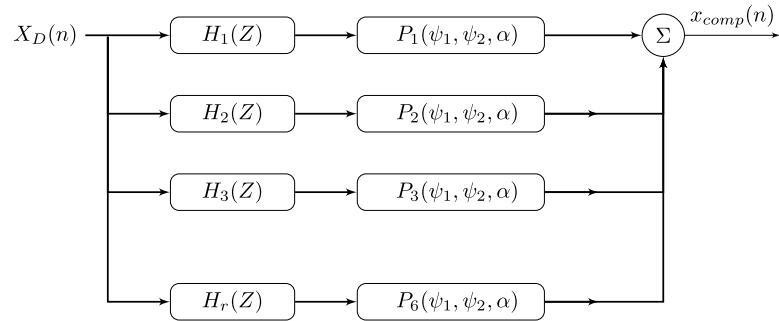
$[\Sigma]_{600 \times 1210}$  = Weight of 1-D sub-filters and 3-D polynomials

$r$  = Rank of matrix equal to number of non-zero singular values

The rank of the matrix  $A$ , is obtained as  $r = 107$ . We have plotted the obtained singular values with respect to its index and shown in Fig.5.2. It may be noted that the variations in singular values is smaller at higher indices (*Index > 4*) when compared with



**Figure 5.2** Index V/s Singular Value plot



**Figure 5.3** Architecture of SVD-VDF based Joint Compensation and Interpolation Filter

the variations in singular values at lower indices. As the low variation in singular values represents noise in the signal, we have selected  $k = 6$  and approximated the matrix  $A$  as:

$$A \approx \sum_{i=1}^6 \sigma_i u_i v_i, \quad (5.19)$$

**Step 5:**

The weighted singular vectors,  $\tilde{u}_i$  are realized as constant 1-D FIR filters. We have designed these FIR filters designed using FIR filter design methods in MATLAB and selected the order of FIR filter as 32. The weighted singular vectors,  $\tilde{v}_i$  are realized by fitting a 3-D polynomial using polyfitn function in MATLAB. We have selected the order of 3-D polynomial as (4, 4, 3) such that the least square error is minimized.

**Step 6:**

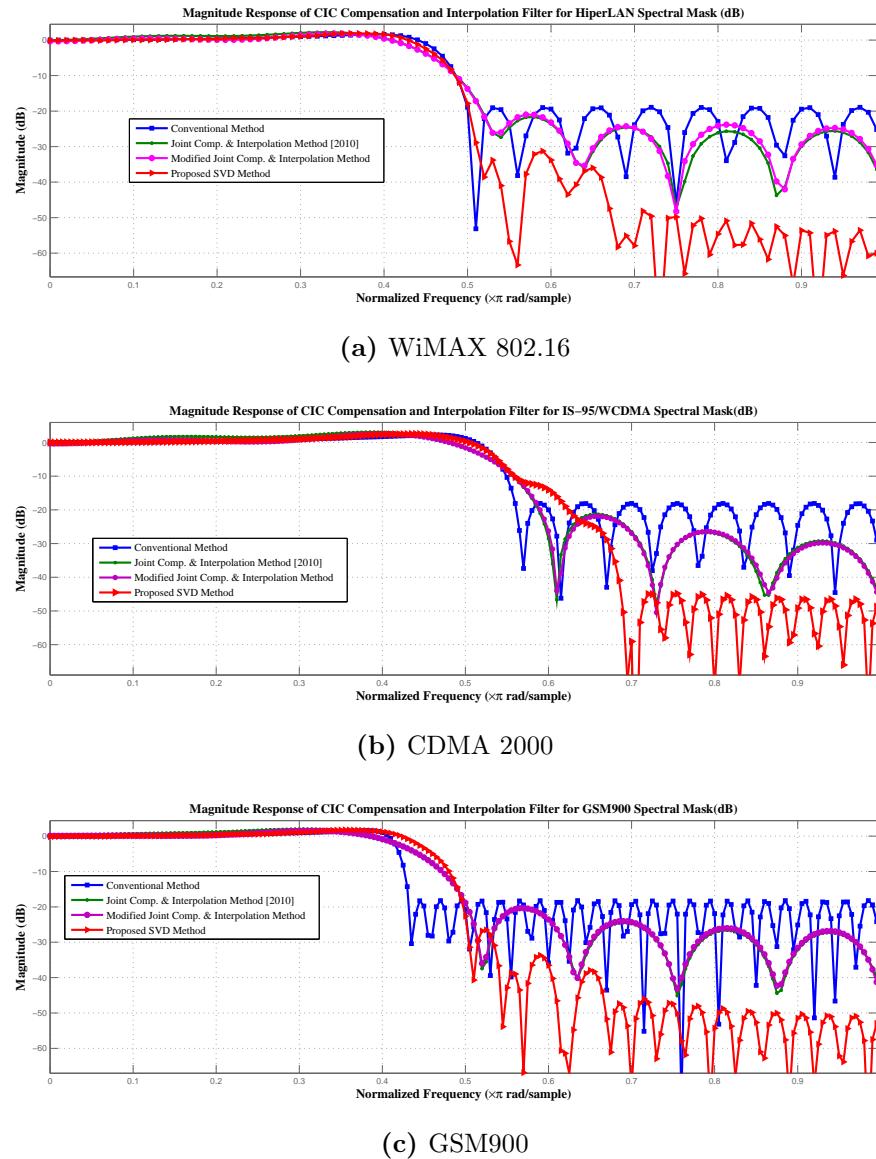
The designed constant 1-D sub-filters are connected with their respective 3-D polynomials to form a VDF for joint compensation and interpolation for three radio standards. Fig.5.3 shows the architecture of a SVD-VDF based reconfigurable joint compensation and interpolation filter with six constant sub-filters connected with their respective 3-D polynomials. Further, we present the performance of the designed filter in terms of frequency response and hardware complexity.

## 5.5 Performance Comparison

In this section, we analyse the performance of the proposed architecture in terms of its frequency response and hardware complexity and compare with the existing methods proposed in the literature.

### 5.5.1 Frequency Response

The proposed SVD-VDF based joint compensation filter is simulated in MATLAB to study its frequency response. We also compare the frequency response of the proposed filter with the frequency response of conventional discrete compensation and interpolation filter, joint compensation and interpolation filter and modified joint compensation and interpolation filter for four radio standards. Fig.5.4 shows the frequency response plot of the discrete compensation filter, joint compensation and interpolation filters as described in Chapter 4 and the proposed SVD-VDF based joint compensation and interpolation filter. The frequency response of a filter is studied by observing its pass band and the stop band characteristics. From the frequency response plot it is observed that the discrete compensation method has sharp spectral characteristics in the pass band region when compared with the other design methods. However, the discrete compensation filter when cascaded with the fractional rate interpolation filter designed based on numerical interpolation techniques results in poor spectral characteristics as stated in section 4.4.1. It may also be observed from section 4.4.2 that the joint compensation and interpolation filter designed based on polynomial approximation of inverse CIC frequency response [85] results in high



**Figure 5.4** Frequency Response of various CIC Compensation and Interpolation Filters

gain errors. Hence, we compare the pass band characteristics of the proposed SVD-VDF based joint compensation and interpolation filter with the modified joint compensation and interpolation filter designed based on frequency domain polynomial approximation and presented in Table 5.2.

**Table 5.2** CIC Gain Compensation achieved with different methods(in dB)

Radio Standard	WiMaX 802.16	CDMA-2000	GSM-900
Required	1.74	2.48	1.64
Joint Comp. [85]	0.6	0.59	1.96
Modified Joint Comp.	0.74	0.83	1.42
Proposed	1.75	2.47	1.68

Since the modified joint compensation and interpolation filter designed based frequency domain polynomials produces frequency warping in the pass band of interest, the gain compensation achieved employing this method deviates by 1dB for WiMAX 802.16, 1.65dB for WCDMA/CDMA2000 and 0.22 dB for GSM 900 radio standards from the required gain compensation. It may also be noted that gain compensation achieved by employing the proposed SVD-VDF based joint compensation and interpolation filter deviates by 0.01dB for WiMAX, 0.01dB for WCDMA/CDMA2000 and 0.04dB for GSM900 radio standards when compared with the ideal gain compensation required. From the frequency response plot, it is observed that the proposed SVD-VDF based filter has better stop band attenuation characteristics when compared with the filter designed using existing methods in the literature. Moreover, the spectral characteristics of the proposed filter are reconfigurable with the spectral parameters. Hence, it can be concluded that the proposed SVD-VDF based joint compensation and interpolation filter produces reconfigurable and better spectral characteristics when compared with the existing methods proposed in the literature.

### 5.5.2 Hardware Complexity

The hardware complexity of the proposed architecture is computed analytically in terms of number of MAC units, so as to compare with the architectures available in the

**Table 5.3** Hardware Complexity Comparison of DDC-SRC based on CIC filters  
(in terms of MAC units)

Method	Compensation FIR	Fractional SRC
Discrete Comp. [81]	$N$	$(L + 1)(L + 1)$
Modified Joint Comp. Method [85]		$L + N + LN + 1$
Proposed Method		$L + N + LN + LP$

literature.

The CIC compensation and interpolation filter designed based on discrete compensation and interpolation method employs  $N$  coefficients for the design of CIC compensation filter resulting in a hardware complexity of  $N$  multiply and accumulate units (MAC). The filter coefficients for attaining the spectral characteristics of various radio standards are programmed through microprocessor based register programming. The symbol rate requirements are attained by employing fractional rate polynomial interpolation filter realized as a Farrow structure. A fractional rate polynomial interpolation filter realized as a  $L^{th}$  order Farrow structure consists of  $(L + 1)$  poly-phase branches with  $(L + 1)$  coefficients in each branch. Hence, this results in a total implementation complexity of  $(N + (L + 1)(L + 1))$  MAC units.

A Farrow structure based Joint compensation and Interpolation method eliminates the need of explicit fractional SRC filter. Since the structure of the modified joint compensation and interpolation filter exactly follows the structure of joint compensation and interpolation filter [85], their hardware complexities are identical. It may be noted that in this structure, a Farrow structure of order  $L$  with  $(L + 1)$  poly-phase branches and  $(N + 1)$  coefficients in each branch results in an implementation complexity of  $(L + N + LN + 1)$  MAC units. It may be noted that the realisation of the above methods for multi-standard radio receivers with  $N_s$  standards requires  $N_s$  look-up tables/offline computations which makes them inflexible to adapt to an existing or forthcoming radio standard.

In contrast to the above methods, the proposed method offers high degree of flexibility to adapt to any existing or forthcoming radio standard spectral mask. The architecture of the filter designed using proposed method is realised as a Farrow structure. A multi-

**Table 5.4** Hardware Complexity Comparison of DDC-SRC based on CIC Filters implemented on Kintex-7 XC7K325t900-2FFG FPGA

Method	#Slices	Reconfiguration
Discrete Comp. Method [81]	23804	Inflexible
Modified Joint Comp. Method [85]	50646	Inflexible
Proposed Method	126218	Flexible

dimensional polynomial in each poly-phase branch of Farrow structure contributes to the variable frequency response. For an  $L^{th}$  order Farrow structure with  $(N + 1)$  coefficients in  $(L + 1)$  branches and  $P$  being the number of coefficients required for the construction of multi-dimensional polynomial in each branch, the hardware complexity of the proposed architecture is computed as  $(L + N + LN + 1)$  and  $LP$  MAC units, respectively.

The comparison of the hardware complexity of the proposed SVD-VDF based joint compensation and interpolation filter architectures with the existing filter architectures presented in the literature are summarized and presented in Table 5.3. It may be noted that, the discrete compensation and interpolation filter require low hardware resources when compared with the joint compensation and interpolation filters. However, they result in poor spectral characteristics and requires offline computations for its reconfiguration. It may also be noted that the joint compensation and interpolation filter designed based on frequency domain polynomial approach improves the spectral characteristics with slight increase in hardware complexity when compared to the discrete compensation and interpolation filter. However, the filter designed employing this approach also could not eliminate the need of offline computations. The proposed SVD-VDF based joint compensation and interpolation filter produces tuneable spectral characteristics at the expense of increase in hardware complexity by  $LP$  MAC units when compared with the existing joint compensation and interpolation filter architecture [85].

The functional simulation of the proposed SVD-VDF based joint compensation and interpolation filter is carried out by developing VHDL models using Xilinx Vivado Design suite software. The proposed SVD-VDF with the design specifications stated in Section 5.4 is implemented on Kintex-7 device family FPGA to estimate its hardware complexity

and compared with the hardware complexity of the DDC-SRC filter architectures designed in Section 4.4 and presented in Table 5.4. It may be observed the proposed SVD-VDF based joint compensation filter architecture consisting of fixed coefficient sub-filters and multi-dimensional polynomials in terms of spectral parameters such as pass band edge frequency, stop band edge frequency and fractional rate interpolation factor produces reconfigurable spectral characteristics with a significant increase in hardware complexity.

## 5.6 Conclusions

In this chapter, the design and implementation of reconfigurable SRC filter to tune the spectral characteristics such as pass band edge frequency, stop band edge frequency and fractional delay as required for multi-standard radio communications is proposed. The designed SVD-VDF based joint compensation and interpolation filter is simulated in MATLAB to observe its frequency response. It is observed that the proposed SVD-VDF filter attains better spectral characteristics when compared with the existing compensation and interpolation filters available in the literature. In addition, the functionality of the proposed filter is verified by implementing it on Kintex-7 FPGA. It is observed that the spectral characteristics of the proposed filter can be easily reconfigured at the expense of significant increase in the hardware complexity when compared to the existing joint compensation and interpolation filter architecture. Hence, an area architecture with reduced hardware complexity to design SVD based reconfigurable SRC filters is proposed in the next chapter.

# Chapter 6

## Area Efficient Architecture for Reconfigurable SRC Filters in SDR Receivers

This chapter presents a multiplier-less architecture proposed in this work to design FIR filters employing Distributed arithmetic (DA). These DA based FIR filters are used to reduce the hardware complexity and latency of the SVD-VDF based reconfigurable SRC filter proposed in Chapter 5. In this chapter ROM-LUT based DA-FIR filter is also proposed to achieve the further reduction in hardware complexity of SVD-VDF based reconfigurable SRC filter. The performance parameters such as hardware complexity and the latency of the proposed ROM-LUT based DA-FIR architectures are computed analytically and compared with the existing DA-FIR architectures available in the literature. The proposed DA-FIR architectures are modeled using VHDL, and the synthesized netlist is implemented using Synopsis Design Vision Compiler with TSMC CMOS 90nm ASIC technology. The ASIC implementation results of the proposed DA-FIR architectures are compared with the DA-FIR architectures available in the literature.

### 6.1 Introduction

SVD-VDF based reconfigurable SRC filter proposed in the Chapter 5 is designed using conventional multiply and accumulate units. The hardware complexity, latency and power consumption can be reduced by employing multiplier-less architectures.

It may be observed that the canonic signed digit (CSD) multipliers is designed using

multiplier-less architecture by expressing the filter coefficients as sum-of-powers-of-two (SOPOT) to reduce the computational complexity. Further reduction in computational complexity can be achieved by eliminating common sub-expressions in the formation of SOPOT coefficients using multiplier block technique [89], [90]. However, the latency remains unaffected by employing these multipliers in the pipelined implementation of FIR filter architecture. It may be noted that the latency of the FIR filters can be reduced by employing distributed arithmetic (DA) based architectures.

DA based FIR filters, [91] are cost-effective and area efficient structures supporting high input sampling rates. The structure of DA-FIR filter consists of LUTs for storing the inner products computed from the filter coefficients. Systolic RAM-LUT based DA-FIR architectures [92]- [94] were proposed in the literature for the realization of filters whose coefficients change during runtime or variable coefficient FIR filters. The hardware complexity of these architectures is high due to more number of RAM-LUTs involved in their realization. Reconfigurable FIR filters can be efficiently realized by optimal sharing of RAM-LUT contents as described in the design of shared RAM-LUT DA-FIR architecture [95]. In this architecture, the inner products stored in RAM-LUTs are shared by employing multiplexers which results in the reduction of the required number of RAM-LUTs and in turn hardware complexity. A block processing shared RAM LUT DA-FIR architecture increases the throughput of the architecture while employing the same number of RAM-LUTs [96]. However, the implementation of RAM based LUT is a costlier effort when compared to ROM-LUTs. Since SVD-VDF filters are fixed coefficient filters the hardware complexity of the reconfigurable SRC filter can be further reduced by employing ROM-based LUTs, resulting in a low latency area efficient architecture.

## 6.2 Distributed Arithmetic Algorithm based FIR Filters

The design methodology for the implementation of FIR filters [94]- [95] employing distributed algorithm with filter order decompostion is presented briefly. The output of an FIR filter,  $y(n)$  with input sequence,  $x(n)$  and impulse response coefficients,  $h(n)$  and filter order  $N$  is

$$y(n) = \sum_{k=0}^{N-1} h(k)x(n - k) \quad (6.1)$$

Replacing the term  $x(n - k)$  in Eq.6.1 with an intermediate term  $w(k)$  for simplification, the above equation can be rewritten as:

$$y(n) = \sum_{k=0}^{N-1} h(k)w(k) \quad (6.2)$$

Assuming the word-length of input sequence as  $L$ , the two's complement representation of  $w(k)$  can be indicated as:

$$w(k) = -[w(k)]_{L-1} + \sum_{l=0}^{L-2} [w(k)]_l 2^{-(L-1-l)} \quad (6.3)$$

where  $[w(k)]_{L-1}$  and  $[w(k)]_l$  denote the index of most significant bit and  $l^{th}$  bit of the input sequence  $w(k)$ , respectively. From Eq.6.2 and Eq.6.3, the FIR filter output can be obtained as:

$$\begin{aligned} y(n) &= \sum_{k=0}^{N-1} h(k) \left( -[w(k)]_{L-1} + \sum_{l=0}^{L-2} [w(k)]_l 2^{-(L-1-l)} \right) \\ &= - \left( \sum_{k=0}^{N-1} h(k) [w(k)]_{L-1} \right) + \sum_{l=0}^{L-2} 2^{-(L-1-l)} \left( \sum_{k=0}^{N-1} h(k) [w(k)]_l \right) \end{aligned} \quad (6.4)$$

It is clear from Eq.6.4 that the output sequence is computed as the shifted summation of  $L$  distributed inner products and expressed as:

$$\begin{aligned} y(n) &= - \left( \sum_{k=0}^{N-1} h(k) [w(k)]_{L-1} \right) + 2^{-1} \left( \sum_{k=0}^{N-1} h(k) [w(k)]_{L-2} \right) + \dots + 2^{-(L-1)} \left( \sum_{k=0}^{N-1} h(k) [w(k)]_0 \right) \\ &= -C_{L-1} + 2^{-1}C_{L-2} + \dots + 2^{-(L-1)}C_0 \\ &= -C_{L-1} + \sum_{l=0}^{L-2} 2^{-(L-1-l)}C_l \end{aligned} \quad (6.5)$$

where,  $C_l$  is an inner partial product expressed as:

$$C_l = \sum_{k=0}^{N-1} h(k) [w(k)]_l \quad (6.6)$$

Eq.6.6 shows that the computation of each inner product  $C_l$  involves summation of  $N$  weighted filter coefficients, where  $N$ , is the order of FIR filter,  $h(k)$  being the filter coefficients, and  $w(k)$  is the weighing binary input of coefficient. As the weight,  $w(k)$  can take a binary value of either '0' or '1',  $C_l$  can have  $2^N$  combinations with a minimum value of zero and  $\sum h(k)$  as a maximum value. All the possible combinations of the inner product  $C_l$  are stored in LUTs and  $[w(k)]_l$  act as the address input to access the weighted

sum of coefficients. The final output is computed as the shifted sum of inner products as given by Eq.6.5. Thus, the computational complexity is reduced to  $(L - 1)$  adders and  $(L * 2^N)$  LUTs to store the weighted sum of filter coefficients are required. In spite of the low computational complexity of DA-based architecture, the order of the filter limits its speed of operation. The number of LUTs required for storing weighted filter coefficients increases exponentially with an increase in the filter order resulting in high memory access time. For higher order filters, the size of LUTs for storing the weighted combinations of inner product,  $C_l$  can be reduced by expressing  $N$  as a composite number,  $N = MP$ , where,  $M$  and  $P$  are two positive integers.

Thus, Eq.6.6 can be expressed as:

$$C_l = \sum_{p=0}^{P-1} \sum_{m=0}^{M-1} h(m + pM)[w(m + pM)]_l \quad (6.7)$$

Eq.6.7 shows that the required number of LUTs for the computation of each inner product is  $(P * 2^M)$  when compared to  $2^N$  (see Eq.6.6). However, the size of LUTs for each inner product is reduced at the expense of increase in computational complexity by  $(P - 1)$  adders. Thus, the overall hardware complexity of the DAFIR filter is  $(LP - 1)$  adders and  $(L * P * 2^M)$  LUTs. This architecture is modified to achieve variable filter characteristics by employing RAMLUTs to generate variable weighted filter coefficients. Hence, the hardware complexity of the variable filter is the sum of  $(L * P * (2^M M - 1))$  adders for coefficient reconfiguration and  $(LP - 1)$  adders for computation of final output, respectively.

Eq.6.5 shows that the number of accumulation operations is directly proportional to the word-length of the input sequence. By transforming the radix of the input from binary to higher radices, such that the word-length of the input sequence is reduced resulting in the reduction of the number of accumulation operations.

$$R = 2^b; b = 1, 2, 3, \dots, L \quad (6.8)$$

The radix transformed input sequence is expressed as:

$$w_b(k) = -[w_b(k)]_{I-1} + \sum_{i=0}^{I-2} [w(k)]_i R^{-(I-1-i)} \quad (6.9)$$

where,  $I = L/b$ . Substituting Eq.6.9 in Eq.6.2 the FIR filter output is computed as:

$$\begin{aligned}
 y(n) &= \sum_{k=0}^{N-1} h(k) \left( -[w_b(k)]_{I-1} + \sum_{i=0}^{I-2} [w_b(k)]_i R^{-(I-1-i)} \right) \\
 &= - \left( \sum_{k=0}^{N-1} h(k) [w_b(k)]_{I-1} \right) + \sum_{i=0}^{I-2} R^{-(I-1-i)} \left( \sum_{k=0}^{N-1} h(k) [w_b(k)]_i \right) \\
 &= - \left( \sum_{k=0}^{N-1} h(k) [w_b(k)]_{I-1} \right) + R^{-1} \left( \sum_{k=0}^{N-1} h(k) [w_b(k)]_{I-2} \right) + \dots + R^{-(I-1)} \left( \sum_{k=0}^{N-1} h(k) [w_b(k)]_0 \right)
 \end{aligned} \tag{6.10}$$

If

$$D_i = \sum_{k=0}^{N-1} h(k) [w_b(k)]_i$$

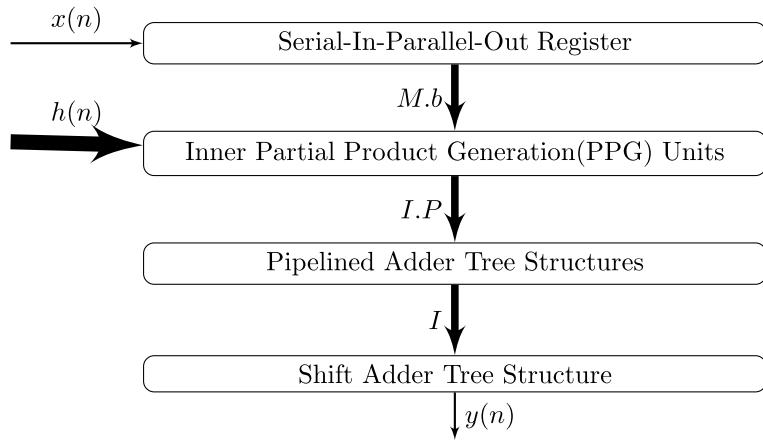
then the filter output  $y(n)$  can be expressed as

$$y(n) = -D_{I-1} + \sum_{i=0}^{I-2} R^{-(I-1-i)} D_i \tag{6.11}$$

The number of accumulation operations gets reduced by a factor of  $b$  as shown in Eq.6.11 when compared to Eq.6.5.

Eq.6.11 shows that each computation of inner product  $D_i$  is the summation of  $N$  weighted filter coefficients, with  $N$ ,  $h(k)$  and  $[w_b(k)]$  is the order of the filter, filter coefficient, and radix transformed weighing input of the filter coefficient, respectively.  $N$  look-up tables of depth  $2^b$  with weighted filter coefficients as their contents are formed. The sum of the required weighted filter coefficients in a pipelined adder tree (PAT) structure computes the inner product, and the shifted-sum of these inner products in a shift adder tree (SAT) structure computes the final filter output.

It may be noted that the total number of LUTs required with the radix transformation is equal to  $(\frac{L}{b} * N)$  with a depth of  $2^b$  LUTs, whereas it is equal to  $(\frac{N}{M} * L)$  with a depth of  $2^M$  LUTs. If  $M = b$  then the number of LUTs required remains same with both the methods. The hardware complexity for computation of final output is proportional to the number of PATs and SATs utilized in the architecture. It may be noted that when the input sequence is transformed to a higher radix, though the number of adders in PATs remain unchanged, the number of stages of PATs and the number of adders in SATs are reduced in the architecture. The number of adders in PATs can be reduced with an increment in  $M$ . Thus, by radix transformation of input from radix-2 to radix- $2^b$  and



**Figure 6.1** Block Diagram of Distributed Arithmetic Based FIR Filter

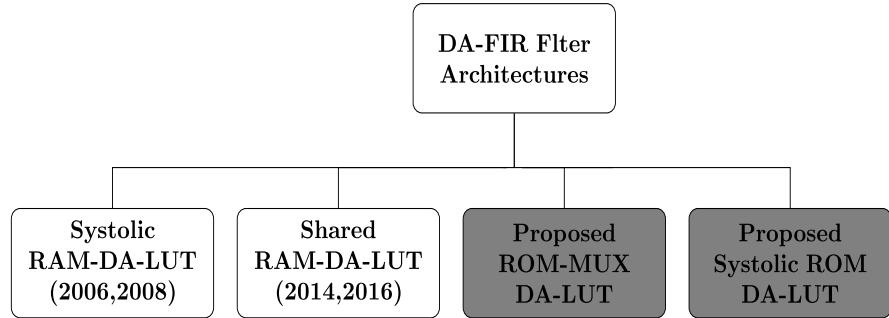
decomposition of filter order as  $N = MP$ , the term  $D_i$  in Eq.6.11 can be expressed as:

$$D_i = \sum_{p=0}^{P-1} \sum_{m=0}^{M-1} h(m + pM) [w_b(m + pM)]_i \quad (6.12)$$

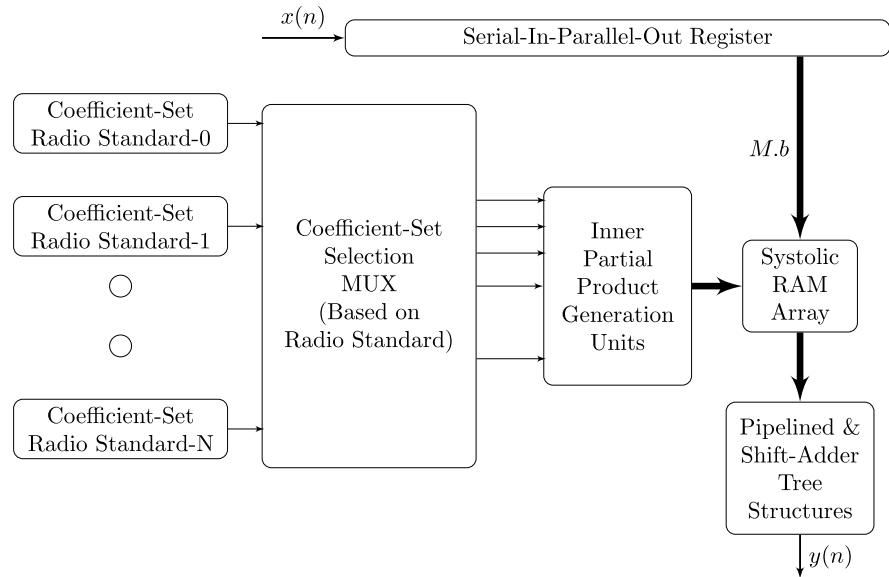
The block diagram for implementation of DA FIR filter based on Eq.6.11 shown in Fig.6.1 is composed of inner partial product generators, PATs, and SATs. The number of PPGs, PAT stages utilized in the architecture is  $\frac{N}{M}$  and  $\frac{L}{b}$ , respectively. The number of PPGs gets reduced with increment in  $M$ , while the increment in  $b$  results in a reduction of the number of PAT stages and adders in SAT stage. The cost incurred is the increased LUT size of PPGs with a proportionate increase in the size of the memory decoding circuit leading to high memory access times. However, an optimum value of  $M$  and  $b$  results in area optimized, high- speed architecture.

### 6.3 Distributed Arithmetic based FIR Filter Architectures

Fig.6.2 shows various distributed arithmetic (DA) based FIR filter architectures proposed in the literature. The performance parameters such as latency, speed and area complexity for each of these architectures are computed analytically and presented in the following sub-sections.



**Figure 6.2** Various DA-FIR Filter Architectures proposed in the literature



**Figure 6.3** Block Diagram of Systolic-RAM DA-FIR Filter Architecture

### 6.3.1 Systolic RAM DA-FIR Filter Architecture

The architecture of Systolic RAM DA-FIR filter architecture [94], [93] is shown in the Fig.6.3. It consists of  $N$ -set of filter coefficients to support  $N$  radio communication standards and a coefficient selection multiplexer to select the coefficient set based on the radio standard into consideration. This selected coefficient set is used to generate systolically all the possible combinations of inner partial products based on filter decomposition factor,  $M$  and word-length decomposition factor,  $b$  by employing adders and stored in the form of a Systolic RAM array. The address lines of the RAM are connected to an appropriate combination of bits of the shifted input sequences for fetching the inner par-

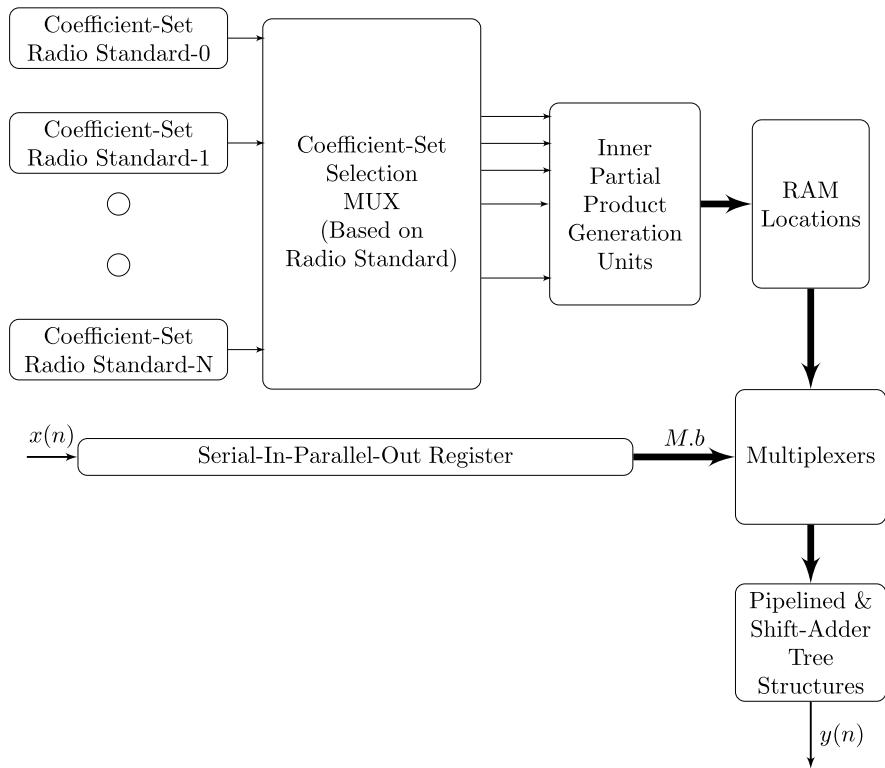
tial products stored in the systolic RAM array. These partial products are accumulated by employing adders in PAT and SAT structures to generate the final filter output. The hardware complexity of DA-FIR filter architecture is the sum of the hardware complexity involved in the generation of inner partial products, storage unit, and accumulation unit. This Systolic RAM DA-FIR filter architecture results in high hardware complexity due to increased number of adders and RAM locations required to generate the inner partial products and to store them, respectively.

### 6.3.2 Shared RAM DA-FIR Filter Architecture

The hardware complexity of the systolic RAM array architecture is reduced by employing shared RAM DA-FIR filter architecture [95], [96]. The shared RAM-LUT DA-FIR filter architecture employs  $N$ -set of filter coefficients to support  $N$  radio communication standards and a coefficient selection multiplexer as described in systolic RAM-DA FIR filter architecture. In this architecture (See Fig.6.4), one set of possible combinations of inner partial products based on filter decomposition factor,  $M$  and word-length decomposition factor,  $b$  are generated from the selected coefficient set and stored in RAM-LUTs. These partial products are shared using multiplexers and the selection lines of the multiplexers are connected to an appropriate combination of bits of shifted input sequences to fetch the partial products. The final output of the filter is computed by accumulating the fetched partial products using adders in PAT and SAT structures. As the hardware complexity of multiplexer array is low when compared to the RAM array, shared RAM-LUT based DA-FIR filter architecture requires low hardware when compared to the systolic RAM DA-FIR filter architecture.

In SDR receivers, since the coefficients of the FIR filter vary with the change in radio communication standard, the inner partial products are generated online and stored in RAM-LUTs leading to increase in hardware complexity. The hardware complexity can be further reduced if the inner partial products are generated offline and stored in ROM-LUTs instead of RAM-LUTs. However, the number of offline computations and ROM-LUTs increase exponentially with increase in the number of radio standards.

SVD-VDF based reconfigurable SRC filter presented in Chapter 5 employs fixed

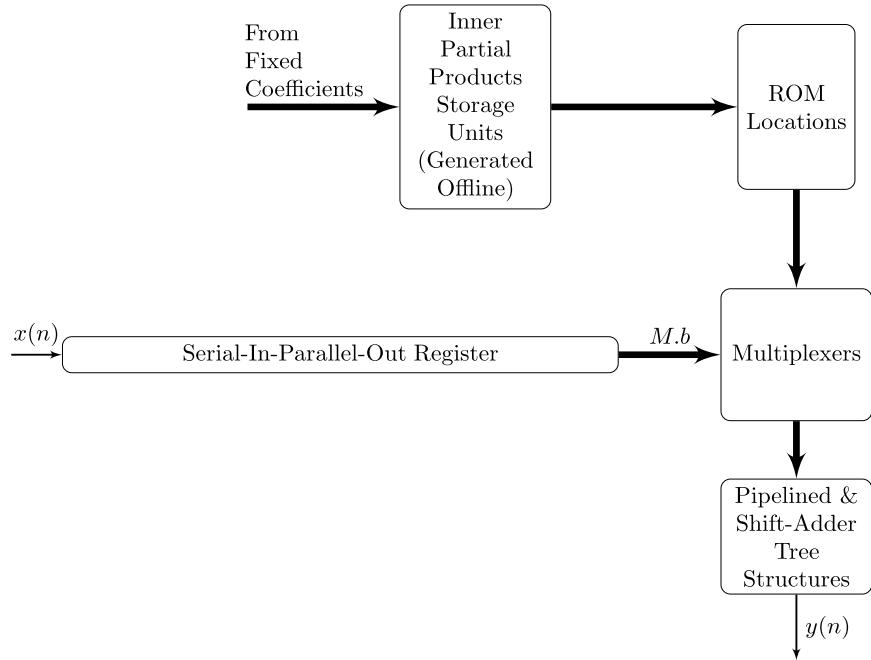


**Figure 6.4** Block Diagram of Shared-RAM DA-FIR Filter Architecture

number of fixed coefficient sub-filters connected to their respective multi-dimensional polynomials defined in terms of variable spectral parameters for supporting multi-standard radio communications in SDR. Hence, the hardware complexity of SVD-VDF is independent of the number of radio communication standards. The hardware complexity of the SVD-VDF can be further reduced by employing ROM-LUT based DA-FIR filter architectures in the design of constant coefficient sub-filters.

### 6.3.3 Proposed ROM-MUX DA-FIR Filter Architecture

Fig.6.5 shows the proposed ROM-MUX architecture with the inner partial products are generated offline from fixed coefficients and stored in ROM-LUTs. These partial products are shared by employing multiplexers as described in shared RAM DA-FIR filter architecture and the final output of the filter is computed by accumulating the inner partial products in PAT and SAT structures. The hardware complexity of the proposed filter is reduced by the number of adders required to organize the contents of DA-LUTs



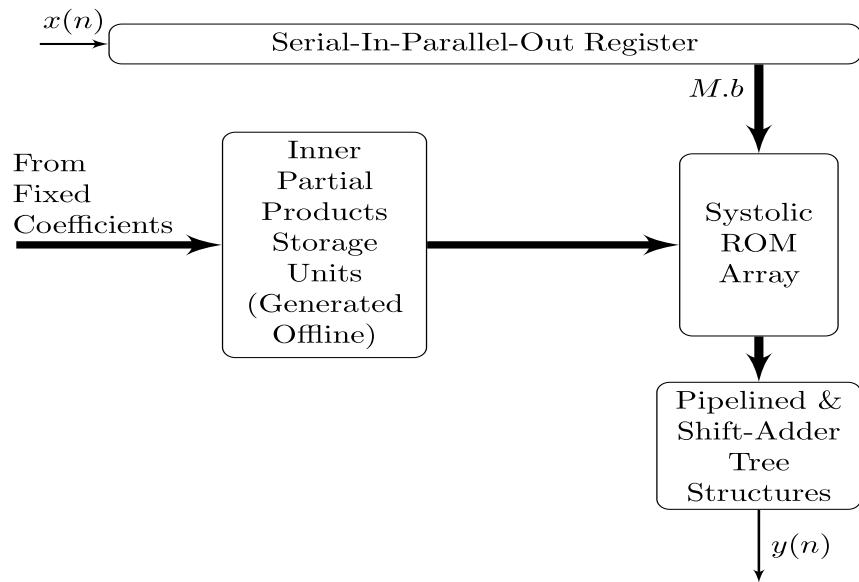
**Figure 6.5** Block Diagram of Proposed ROM-MUX DA-FIR Filter Architecture

in shared RAM DA-FIR filter architecture. However, it is observed that the hardware complexity of an  $n$ -bit ROM is low when compared to the hardware complexity of an  $n$ -bit multiplexer [20]. Hence, the hardware complexity of the DA-FIR filter architecture can be further reduced by replacing multiplexers with ROM-LUTs and arrange them in the form of a systolic array.

#### 6.3.4 Proposed Systolic ROM DA-FIR Filter Architecture

In this architecture, the inner partial products are generated offline and stored in ROM-LUTs arranged in the form of a systolic array as shown in Fig. 6.6. The address lines of the ROM are connected to an appropriate combination of bits of shifted input sequences to fetch the inner partial products stored in the systolic ROM array. The final output of the filter is computed by accumulating these partial products in the PAT and SAT structures. It may be noted that the proposed systolic ROM DA-FIR filter architecture has low hardware complexity when compared to the ROM-MUX DA-FIR filter architecture.

The next section presents the performance analysis in terms of hardware complexity



**Figure 6.6** Block Diagram of Proposed Systolic ROM DA-FIR Filter Architecture

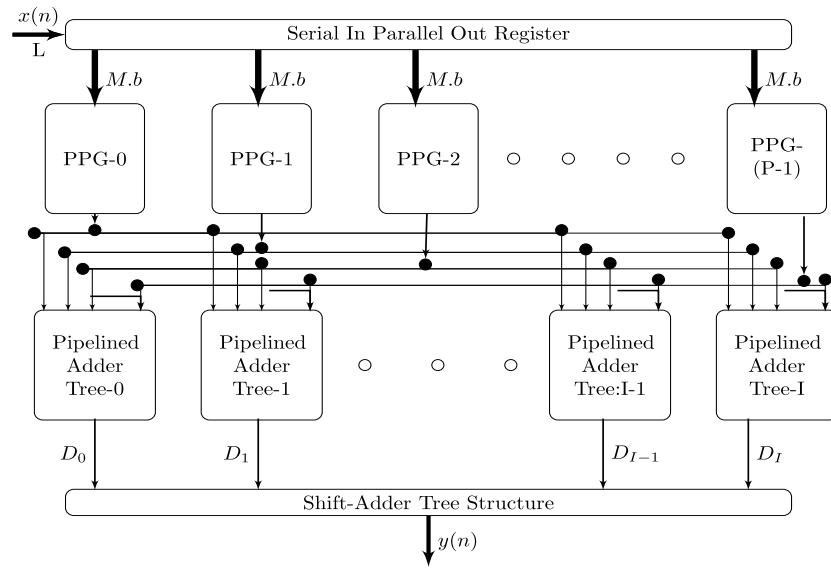
and time complexity of the various DA-FIR filter architectures presented in this section.

## 6.4 Analysis of DA-FIR Filter Architectures

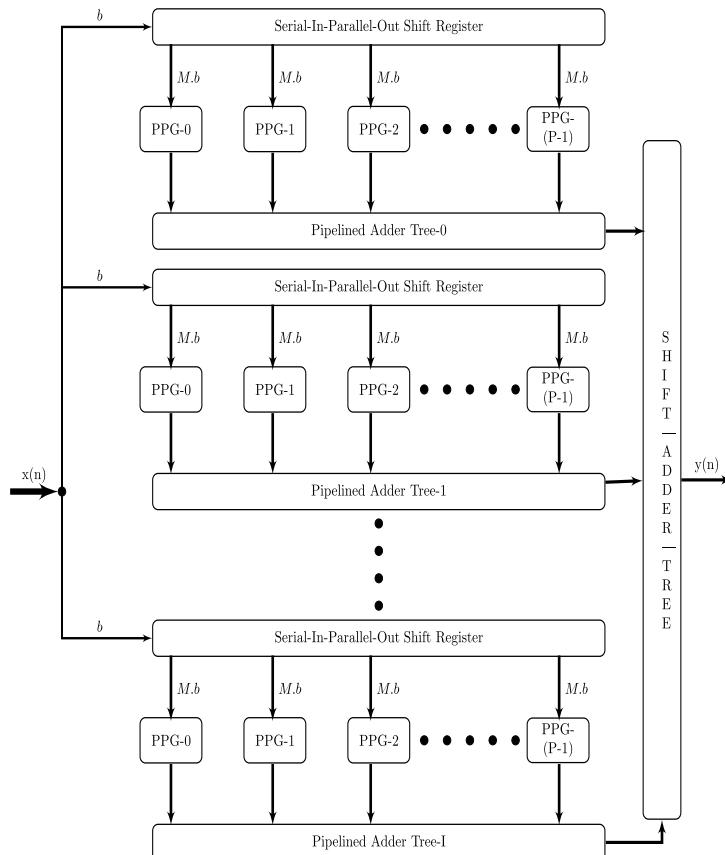
In this section, the hardware complexity and the time complexity of the proposed ROM-LUT based DA-FIR filter architectures are analyzed and compared with the existing shared RAM-LUT DA-FIR filter architecture available in the literature.

### 6.4.1 Hardware Complexity

The hardware complexity of the DA-FIR filter architecture is computed as the sum of hardware complexity involved in the organization of DA-LUTs and accumulation of partial products. Fig. 6.7a shows the DA-FIR filter architecture which employs sharing of partial products stored in LUTs. The architecture consists of  $P$  partial product generators (PPGs) requiring  $P$  LUTs to store these inner partial products and shared using  $(P * I)$  multiplexers. Fig. 6.7b shows the systolic architecture for DA-FIR filter employing  $(P * I)$  partial product generators and requires  $(P * I)$  memory LUTs to store these inner partial products. The number of adders employed for accumulation of partial products remains



(a) Shared Memory DA-FIR Filter Architecture



(b) Systolic Array DA-FIR Filter Architecture

**Figure 6.7** Functional diagram of DA-FIR Filter Architectures

same in shared memory architecture and systolic memory array architecture. Hence, it may be noted that the hardware complexity of DA-FIR filter architecture mainly varies with the hardware complexity required in organizing DA-LUTs and fetching the inner partial products from them. The hardware complexity of the DA-FIR filter architectures presented in Section 6.3 is analyzed.

In RAM-LUT based DA-FIR filter architectures, the PPGs consist of adders and RAM-LUTs to organize the contents of DA-LUTs. With  $N$ ,  $M$  and  $b$  being the filter order, filter order decomposition factor and word-length decomposition factor, respectively, the depth of each RAM required is  $(2^{M*b})$  words. The number of adders required to organize the DA-LUT contents in the RAM locations is  $(2^{M*b} - M * b - 1)$ . In ROM-LUT based DA-FIR filter architectures, RAM-LUTs are replaced by ROM-LUTs and the DA-LUT contents are computed offline.

Systolic RAM DA-FIR filter architecture employs  $(P * I)$  PPGs. Hence, the hardware complexity involved in the organization of DA-LUT contents in systolic RAM DA-FIR filter architecture is  $(P * I * (2^{M*b} - M * b - 1))$  adders and  $(P * I * (2^{M*b}))$  word RAM-LUTs.

Shared RAM DA-FIR filter architecture employs  $P$  PPGs with  $P * I$  multiplexers. The size of each multiplexer required is  $(2^{M*b} - to - 1)$ . The hardware complexity required for DA-LUT organization in the shared RAM DA-FIR filter architecture is computed as  $(P * (2^{M*b} - M * b - 1))$  adders,  $(P * (2^{M*b}))$  word RAM-LUTs and  $P * I * (2^{M*b} - to - 1)$  multiplexers.

In the ROM-MUX DA-FIR filter architecture, the hardware complexity required for DA-LUT organization is computed as the sum of the hardware complexity of  $(P * (2^{M*b}))$

**Table 6.1** Hardware Complexity of DA-FIR Filter Architectures

Architecture	#Adders	#RAM	#ROM	#MUX(2:1)
Systolic RAM [94], [93]	$P * (2^{M*b} - M * b - 1)$	$P * I * (2^{M*b} - 1)$	0	0
Shared RAM [95], [96]	$P * (2^{M*b} - M * b - 1)$	$P * (2^{M*b} - 1)$	0	$P * I * (2^{M*b}) - 1$
Proposed-I	Offline	0	$P * (2^{M*b})$	$P * I * (2^{M*b}) - 1$
Proposed-II	Offline	0	$I * P * (2^{M*b})$	0

ROM-LUT words and hardware complexity of  $(P * I * (2^{M*b} - to - 1))$  multiplexers. Since, the DA-LUT contents are computed offline the hardware complexity is reduced by the number of adders required to organize the DA-LUTs.

The hardware complexity required for DA-LUT organization in the systolic ROM DA-FIR filter architecture is computed as  $(I * P)$  ROM-LUTs with a depth of  $(2^{M*b})$  words. Since the hardware complexity of  $(2^{M*b})$  ROM is low when compared to  $(2^{M*b} - to - 1)$  multiplexer, the hardware complexity of systolic ROM DA-FIR filter architecture is low when compared to the ROM-MUX DA-FIR filter architecture.

The hardware complexity for the organization of DA-LUT contents of various DA-FIR filter architectures is summarized in Table 6.1. In addition to the hardware required for the generation of these partial products, pipelined adder tree and shift adder tree structures are required to compute the final output of the filter. The hardware complexity of these adder structures can be computed as

$$N_{adders} = \lceil I * \lceil P - 1 \rceil + I - 1 \rceil \quad (6.13)$$

where,  $P = \frac{N}{M}$ ;  $I = \frac{L}{b}$ .

#### 6.4.2 Time complexity

The time complexity of an architecture can be evaluated in terms of speed and latency. The speed of the architecture depends on the longest combinational data path in the architecture. Let the time required to access a filter coefficient from the stored coefficient memory be  $T_{ROM}$ , the maximum adder delay required to compute the contents of LUTs be  $T_A$  and let  $T_{MUX}$  be the MUX delay of 2-to-1 multiplexer. Let the propagation delay of one  $N$ -bit adder be  $T_{Acc}$ . The longest data path in DAFIR architecture is defined as the path from ROMLUTs to the final filter output. Appropriate selection of  $M$  and  $b$  reduces the combinational path delay. This is due to the fact that the number of multiplexing and adder stages varies in accordance with these two parameters. The number of multiplexing stages is given as  $M.b$ , adder stages in PAT and SAT is computed as  $\log_2 P$  and  $\log_2 I$ , respectively. Thus, the required minimum clock period or critical path (CP) can be expressed as:

$$T_{CP,2} = T_{ROM} + T_A + M.b.T_{MUX} + (\log_2 P + \log_2 I)T_{ACC} \quad (6.14)$$

In Eq.6.14, the first two terms are independent of filter order and word length decomposition and the third term contributes to a small delay as  $T_{MUX}$  is very small in comparison to the  $n$ -bit adder delay. When the DA-FIR architecture is implemented using pipelined technique, it may be observed that the latency reduces by one adder stage and the multiplexer stage increases by one for two-fold increase  $M.b$ . However, contribution of the multiplexer delay is very small in comparison to the delay produced by the ripple carry adder. Hence, it can be concluded that in a pipelined architecture reduction in one stage of accumulation reduces the latency of the architecture by one, though the multiplexer stage gets increased. Further, we propose to implement the short-length adders in PAT stage as four input adder trees to reduce the latency of the architecture with a slight increment in delay in the combinational data path. Hence, the required minimum clock period or critical path is now expressed as:

$$T_{CP,4} = T_{ROM} + T_A + M.b.T_{MUX} + (\log_4 P + \log_4 I)T_{ACC} \quad (6.15)$$

**Table 6.2** Computation of Critical Path and Latency of DA-FIR Filter Architectures

Architecture	Critical Path	Latency
Non-SVD Filter	$T_{ROM} + T_A + M.b.T_{MUX} + \phi_1 * T_{ACC}$	$\phi_1 + 3$
Systolic RAM [94], [93]	$T_A + M.b.T_{MUX} + \phi_1 * T_{ACC}$	$\phi_1 + 2$
Shared RAM [95], [96]	$T_A + M.b.T_{MUX} + \phi_1 * T_{ACC}$	$\phi_1 + 2$
Proposed-I	$M.b.T_{MUX} + \phi_1 * T_{ACC}$	$\phi_1 + 1$
Proposed-II	$M.b.T_{MUX} + \phi_1 * T_{ACC}$	$\phi_1 + 1$

$$\phi_1 = \log_4 P + \log_4 I$$

The critical path for the various DA-FIR architectures with an application to design SVD-VDF are computed and presented in Table 6.2. It may be noted that in a non-SVD-VDF filter, the filter coefficients vary with the change in radio communication standard. Hence, these filters have the longest critical path when compared to the SVD based variable digital filters. It is observed from Table 6.2 that the critical path of the RAM-LUT based DA-FIR filter architectures to design SVD-VDF is reduced by the time required to fetch coefficients with the changing radio communication standards (See Row 2 & 3 in

Table 6.2). In ROM-LUT based DA-FIR filter architectures, since the DA-LUT contents are computed offline the computational complexity gets reduced and in turn the critical path gets reduced by an amount of adder delay (See Row 4 & 5 in Table 6.2). Hence, the proposed ROM-LUT based DA-FIR architectures has the shortest critical path when compared to the other architectures proposed in the literature. Further, we propose to implement the DA-FIR filters presented in Table 6.2 as a pipelined architecture and it is observed that the latency of the proposed ROM-LUT based DA-FIR filter architectures improves by one clock cycle when compared with the RAM-LUT based DA-FIR filter architectures.

**Table 6.3** Comparison of Hardware Complexity of DA-FIR Filter

with  $N = 64$ (65-Tap),  $L = 16$ -bit,  $M = 4$ ,  $b = 1$ (Radix-2)

Architecture	Adders LUT-org	Memory/ Register Loc.	MUXERs/ Decoders	Adders PATs & SATs
Systolic RAM [94], [93]	50688	69408 RAM	0	5480
Shared RAM [95], [96]	3168	3858 Registers	69408 Muxers	5480
Proposed-I	0	3858 ROM	69408 Muxers	5480
Proposed-II	0	69408 ROM	12560 Decoders	5480

#### 6.4.3 Performance analysis

In this sub-section, we analyze the hardware complexity of various DA-FIR filter architectures for various filter orders, filter order decomposition factors and word-length decomposition factors. In our work we have selected  $M$  and  $b$  such that their product is constant and equal to 4. First we consider the order of the filter as  $N = 64$  (65-tap filter), filter order decomposition factor,  $M = 4$ , word length,  $L = 16$  and word-length decomposition factor,  $b = 1$ . The hardware complexity of the various DA-FIR filter

architectures with these specifications is evaluated and presented in Table 6.3.

**Table 6.4** Hardware Complexity of DA-FIR Filter Architectures for different Filter Orders, 16-bit Word-length and  $M.b = 4$  in terms of Area of Full adder

(a)  $M = 4, b = 1$  (Radix-2) Architecture

Architecture	$N =$	65-Tap	33-Tap	17-Tap	9-Tap
Systolic RAM [94], [93]		121848	61097	30714	15516
Shared RAM [95], [96]		34803	17576	8954	4636
Proposed-I		28295	14315	7318	3812
Proposed-II		10411	5685	3008	1662

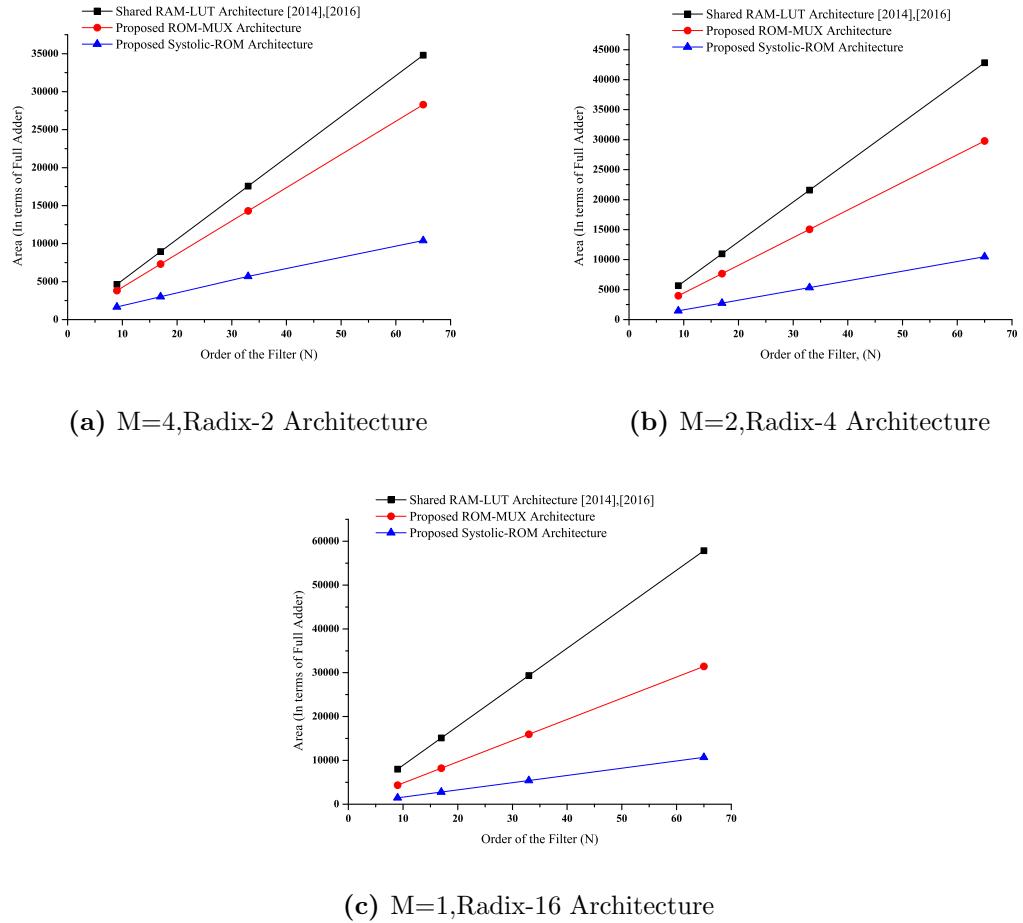
(b)  $M = 2, b = 2$  (Radix-4) Architecture

Architecture	$N =$	65-Tap	33-Tap	17-Tap	9-Tap
Systolic RAM [94], [93]		131770	66259	33500	17117
Shared RAM [95], [96]		42821	21591	10973	5661
Proposed-I		29772	15038	7668	3979
Proposed-II		10486	5335	2755	1462

(c)  $M = 1, b = 4$  (Radix-16) Architecture

Architecture	$N =$	65-Tap	33-Tap	17-Tap	9-Tap
Systolic RAM [94], [93]		137004	69545	35814	18947
Shared RAM [95], [96]		57834	29351	15108	7985
Proposed-I		31444	15953	8206	4331
Proposed-II		10696	5420	2780	1458

The hardware complexity of various DA-FIR filter architectures is computed by expressing the area complexity of various components in terms of area of a full adder. The hardware complexity of a 1-bit register, 2-to-1 MUXER and 2-to-1 decoder in terms of area of a full adder is given as  $0.62A_{FA}$ ,  $0.33A_{FA}$  and  $0.15A_{FA}$ , respectively [20]. The hardware complexity of a RAM is equal to the sum of hardware complexity of a register and multiplexer while that of a ROM is equal to the sum of the hardware complexity of its address decoder and the number of transistors required for placing the required data in a



**Figure 6.8** Order of the Filter versus Hardware Complexity (in terms of area of Full adder)

ROM location. Hence, the hardware complexity of a RAM and ROM location is assumed as  $0.95A_{FA}$  and  $0.03A_{FA}$ , respectively. Table 6.4 presents the hardware complexity of the various DA-FIR filter architectures for different filter orders in terms of area of full adder. Further, the hardware complexity of the DA-FIR filter architectures with  $b = 1$  (Radix-2),  $b = 2$  (Radix-4) and  $b = 4$  (Radix-16) are presented in Table 6.4a, Table 6.4b and Table 6.4c, respectively. It is observed that the Systolic RAM-LUT DA-FIR filter architecture requires more hardware compared to the other DA-FIR filter architectures. It is also observed that the hardware complexity of the proposed ROM-LUT based DA-FIR filter architectures is low when compared to the existing DA-FIR filter architectures available in the literature.

The variation of hardware complexity with variation in filter order for various DA-FIR filter architectures with  $b = 1$  (Radix-2),  $b = 2$  (Radix-4) and  $b = 4$  (Radix-16) input

**Table 6.5** Percentage Reduction in Hardware complexity of Proposed ROM-LUT based DA-FIR Filter Architectures

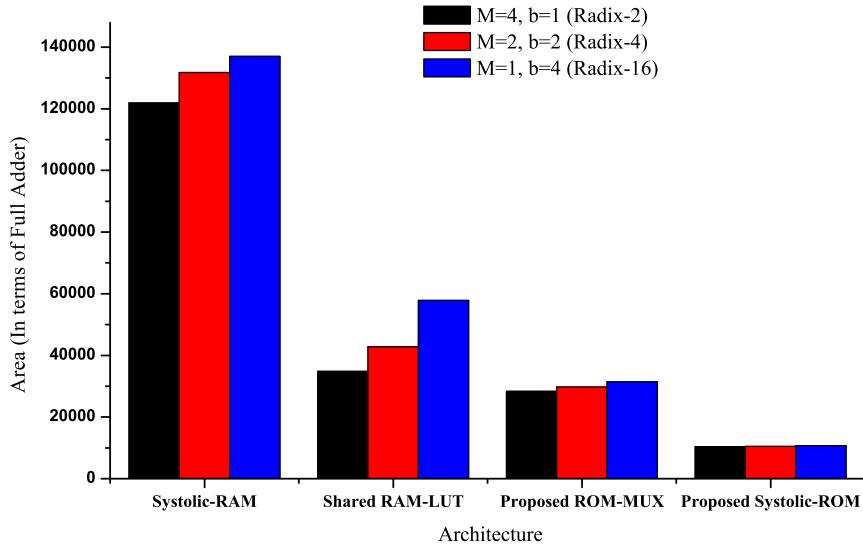
Architecture	Area	%Reduction
Radix-2,M=4		
Shared RAM-LUT [95], [96]	34803	-
Proposed-I ROM-MUX	28295	18%
Proposed-II Systolic ROM	10411	70%
Radix-4,M=2		
Shared RAM-LUT [95], [96]	42821	-
Proposed-I ROM-MUX	29772	30%
Proposed-II Systolic ROM	10486	75%
Radix-16,M=1		
Shared RAM-LUT [95], [96]	57834	-
Proposed-I ROM-MUX	31444	45%
Proposed-II Systolic ROM	10696	81%

sequence is plotted and shown in Fig.6.8.a, Fig.6.8.b and Fig.6.8.c, respectively. The percentage reduction in the hardware complexity of the proposed ROM-MUX and Systolic ROM DA-FIR filter architectures with respect to the hardware complexity of the shared RAM-LUT DA-FIR filter architecture for various input radices is computed and tabulated in Table 6.5. It is observed that the hardware complexity of the proposed ROM-MUX architecture reduces by 18%, 30% and 45% and the systolic ROM architecture reduces by 70%, 75% and 81% for radix-2, radix-4 and radix-16 input sequence, respectively, when compared with the shared RAM-LUT architecture. It is observed that for a constant product of  $M.b = 4$ , the number of computation units and memory locations required for the organization of DA-LUT contents increases with increase in the radix of the input sequence in the shared LUT DA-FIR filter architectures (See Row 3 & Row 4 of Table 6.1,  $P = \frac{N}{M}$ ), while the other hardware units remain consistent. In ROM-LUT based DA-FIR filter architectures with increment in radix of the input sequence, higher number of computation units are replaced by offline computations resulting in high percentage reduction in hardware complexity when compared with the shared RAM-LUT DA-FIR

**Table 6.6** Hardware Complexity and Latency Comparison of DA-FIR Filter with  $N = 64$ (65-Tap) and  $L = 16$ -bit

Architecture	Systolic RAM	Shared RAM	Latency	Proposed-I	Proposed-II	Latency
$M = 4, b = 1$	121840	34803	9	28295	10411	8
$M = 2, b = 2$	131770	42821	8	29772	10486	7
$M = 1, b = 4$	137004	57834	8	31444	10696	7

architecture (See Table 6.5).



**Figure 6.9** Histogram plot for Architecture versus Area (in terms of area of Full adder)

Finally, we compare the hardware complexity and latency of the various DA-FIR filter architectures with variation in radix of the input sequence and presented in Table 6.6. We consider the order of filter as  $N = 64$  (65-Tap), word-length of the input sequence as  $L = 16$ -bit and radix-2 ( $b = 1$ ), radix-4 ( $b = 2$ ) and radix-16 ( $b = 4$ ) for input sequence. It is observed that the hardware complexity of the DA-FIR filter architecture increases with increment in radix of the input sequence. This is due to the required word-length for organization of DA-LUT contents increases with increment in radix. Fig.6.9 shows the histogram plot of variation in hardware complexity with variation in radix of the input sequence. It is observed that the variation in hardware complexity with variation in radix

of the input sequence is very high in Shared RAM-LUT architecture when compared with the other DA-FIR filter architectures. This is due to increment in the computational complexity for the DA-LUT organization with decrement in filter order decomposition factor,  $M$ . Since the contents of DA-LUTs are computed offline in ROM-LUT based DA-FIR architectures and the variation in area complexity of the ROM is low with increment in word-length, it is observed that the proposed ROM-MUX and systolic ROM DA-FIR filter architectures have comparable hardware complexity with variation in radix of the input sequence. It is also observed that the latency of the DA-FIR filter architecture with Radix-4 and Radix-16 input sequence improves by one clock cycle when compared with the Radix-2 DA-FIR filter architecture. Hence, for a constant product of filter order decomposition and word-length decomposition factor of four, the Systolic ROM DA-FIR filter architecture with  $M = 2$ ,  $b = 2$  is an optimum architecture with low latency and optimum area. The performance of the DA-FIR filter architectures for realization of SVD-VDF filter are also compared by implementing on hardware using FPGA and ASIC technologies and presented in next section.

## 6.5 Hardware Implementation

The DA-FIR filter architectures presented in Section 6.4 is applied to design SVD-VDF based reconfigurable SRC filter. We have considered the order of sub-filters in the design of SVD-VDF as Type-1 FIR filter with order,  $N=64$  (65-tap) and word-length of the input sequence as,  $L=16$ -bit. The performance of SVD-VDF with various DA-FIR filter architectures is compared by implementing on FPGA and synthesized using Synopsis design compiler with TSMC CMOS 90nm technology. The DA-FIR based SVD-VDF functionality is verified by developing VHDL models and simulating using Xilinx Vivado 2014.2 software. This VHDL netlist is synthesized and implemented on Kintex-7 XC7K325t900-2ffg FPGA. The performance comparison of RAM-LUT based DA-FIR filter architectures implemented on FPGA are presented in Table 6.7. It is observed that the Systolic RAM DA-FIR based SVD-VDF require more hardware when compared with the shared RAM-LUT based SVD-VDF architecture. It may also be noted that the DA architecture with decomposition factors  $M = 4, b = 1$  occupies low area when compared

**Table 6.7** Performance Comparison of DA-FIR SVD-based SRC Filter implemented on Kintex-7 XC7K325t900-2ffg FPGA

Architecture	#Slices	MSP	Latency
Systolic RAM Array Architecture [94], [93]			
$M = 4, b = 1$	350888	2.719	9
$M = 2, b = 2$	499516	2.77	8
$M = 1, b = 4$	527840	2.818	8
Shared RAM-LUT Architecture [95], [96]			
$M = 4, b = 1$	220552	2.719	9
$M = 2, b = 2$	229714	2.77	8
$M = 1, b = 4$	255815	2.818	8

to the DA-FIR filter architectures with  $M = 2, b = 2$  and  $M = 1, b = 4$ . It is observed that the minimum sampling period (MSP) of the DA-FIR filter architecture increases with increment in  $b$ . This is due to increase in width of the input word-length with the increase in  $b$ . It is also observed that the latency of DA-FIR filter architectures with higher radix input sequence is improved by one clock cycle when compared with DA-FIR filter architecture with radix-2 input sequence. Hence, it can be concluded that the DA-FIR filter architecture with  $M = 2, b = 2$  is an optimum architecture with low latency and optimum area.

VHDL models of distributed arithmetic based architectures presented in Section 6.4 are also synthesized using Synopsis Design Compiler with CMOS 90nm Taiwan Semiconductor manufacturing Company (TSMC) library with the environment variables presented in Appendix A. The hardware complexity, latency and energy per sample (EPS) are tabulated in Table 6.8. Fig.6.10a and 6.10b shows the histogram plot of hardware complexity in terms of cell area (in  $Sq.\mu m$ ) and energy-per-sample (in Watts-ns) for various architectures. It is observed that the hardware complexity of various DA-FIR architectures when synthesized using Synopsis tools are in coherence with the analytical computation of hardware complexity (See Fig.6.9 & Fig.6.10a). It may also be noted that systolic RAM DA-FIR architectures consume more power when compared with the other DA-FIR SVD based SRC filters (See Fig.6.10b). It is also observed that the minimum

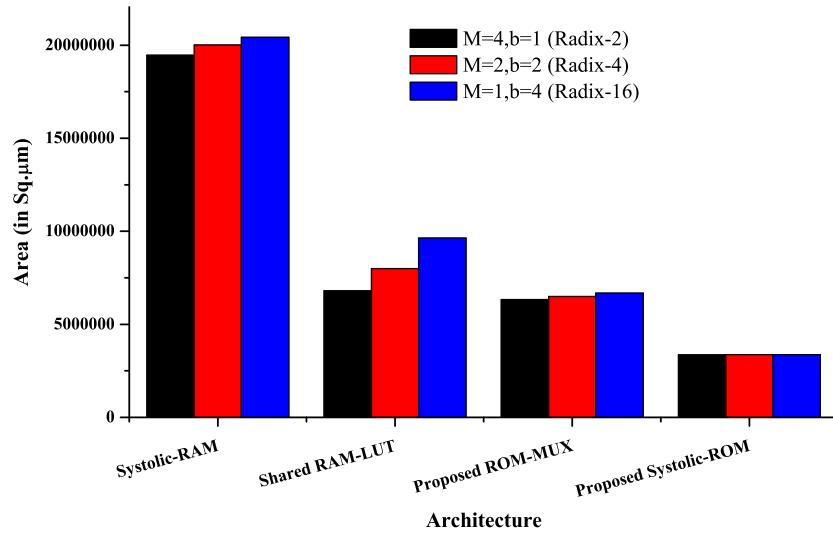
**Table 6.8** ASIC Synthesis Results of DA-FIR SVD-based SRC Filter with  
TSMC CMOS 90nm Library

Architecture	Area	MSP	Power	EPS	Latency
Systolic RAM Array Architecture [94], [93]					
$M = 4, b = 1$	19462808	2.41	3.862	9.307	9
$M = 2, b = 2$	20003672	2.61	4.036	10.53	8
$M = 1, b = 4$	20426836	2.66	3.740	9.948	8
Shared RAM-LUT Architecture [95], [96]					
$M = 4, b = 1$	6801264	2.41	0.773	1.863	9
$M = 2, b = 2$	7998800	2.61	1.254	3.273	8
$M = 1, b = 4$	9634832	2.66	1.553	4.126	8
Proposed-I ROM-MUX Architecture					
$M = 4, b = 1$	6335712	2.41	0.653	1.537	8
$M = 2, b = 2$	6502632	2.61	0.589	1.537	7
$M = 1, b = 4$	6671152	2.66	0.614	1.602	7
Proposed-II Systolic ROM Architecture					
$M = 4, b = 1$	3362940	2.41	0.477	1.15	8
$M = 2, b = 2$	3367036	2.61	0.491	1.28	7
$M = 1, b = 4$	3370228	2.66	0.493	1.31	7

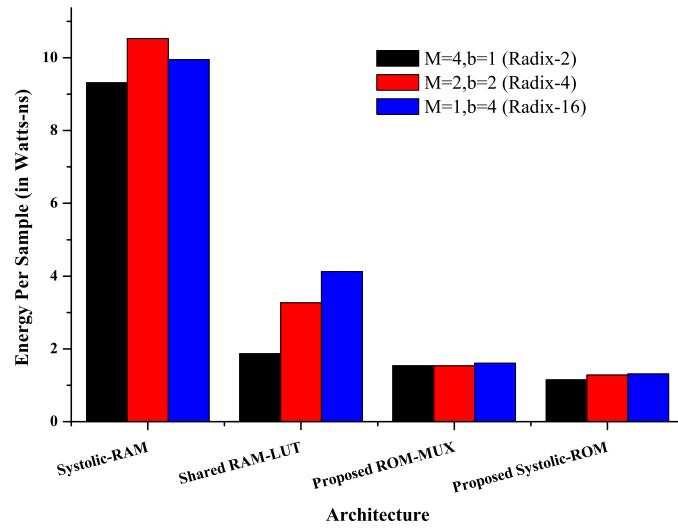
**Units:** Area: Sq. $\mu$ m; Power: Watts(W); MSP: nS; EPS:W-nS

sampling period (MSP) of architectures with higher radix input sequence is slightly high due to increase in word-length of the DA-LUT contents. However, the latency is improved by one clock cycle when compared with radix-2 architecture.

It may also be inferred that among Shared RAM-LUT architectures, the architecture with filter order decomposition factor,  $M = 4$  and radix-2 architecture is efficient in terms of area and energy per sample (EPS). The performance of the proposed ROM-LUT based DA-FIR filter architectures is compared with this Shared RAM-LUT architecture and presented in Table 6.9. It may be noted that the area of the proposed ROM-MUX architecture is comparable to the area of the Shared RAM-LUT architecture with energy



(a) Architecture Versus Cell Area



(b) Architecture Versus Energy per Sample

**Figure 6.10** ASIC Synthesis Results using TSMC CMOS 90nm library

efficiency of more than 14% and latency improvement by one clock cycle. From Table 6.9 it is also inferred that the proposed Systolic ROM architecture achieves hardware reduction by 50% and energy efficiency by more than 30% when compared with the shared RAM-LUT architecture proposed in the literature. It may be noted that among the various Systolic ROM DA-FIR architectures Radix-4 architecture with  $M = 2$  is an optimum

**Table 6.9** Performance Comparison of ASIC Synthesis Results of Proposed DA-FIR SVD-based SRC filter

Architecture	Area	% Reduction	EPS	% Efficiency	Latency
Shared RAM-LUT Architecture [95], [96]					
$M = 4, b = 1$	6801264	-	1.863	-	9
Proposed-I ROM-MUX Architecture					
$M = 4, b = 1$	6335712	6.85	1.537	17.5	8
$M = 2, b = 2$	6502632	4.4	1.537	17.5	7
$M = 1, b = 4$	6671152	1.92	1.602	14.01	7
Proposed-II Systolic ROM Architecture					
$M = 4, b = 1$	3362940	50.56	1.15	38.28	8
$M = 2, b = 2$	3367036	50.5	1.28	31.3	7
$M = 1, b = 4$	3370228	50.44	1.31	29.69	7

**Units:** Area: Sq. $\mu$ m; EPS:W-nS

architecture with low latency, comparable area, and energy efficiency when compared to the Radix-2 architecture with  $M = 4$ . Hence, a low latency area efficient architecture for the design of fixed coefficient SVD VDF based SRC filters is proposed.

## 6.6 Conclusions

In this chapter, an area efficient architecture to design reconfigurable SVD-VDF based SRC filters for SDR receivers by employing distributed arithmetic is presented. Since the coefficients of filters in SVD-VDF based are fixed coefficient filters, the hardware complexity of DA-FIR filter architecture is reduced by proposing ROM-LUT based DA-FIR filter architectures. It is observed from the analytical computations that the proposed systolic ROM-LUT based DA-FIR architecture improves latency by two clock cycles and achieves area efficiency by 70% when compared with the existing shared RAM-LUT DA-FIR architecture. DA FIR architectures for SRC in SDR receivers are modelled in VHDL and synthesized using Synopsis design compiler with TSMC CMOS 90nm library. From

the comparison of ASIC implementation results, it is concluded that the proposed Systolic ROM architecture achieves hardware reduction by 50% and energy efficiency by more than 30% when compared with the shared RAM-LUT architecture proposed in the literature. Hence, it can be concluded that the Systolic ROM architecture proposed in this chapter for the design of SVD-VDF based SRC filter results in reduction in latency and area. The next chapter presents the conclusions drawn from the earlier chapters and the future scope of the research work.

# Chapter 7

## Conclusions and Future Scope

This chapter concludes the thesis by underlining the main contributions and directions to the future scope for the research work.

### 7.1 Conclusions

Recent advancements in wireless technology led to tremendous growth in the wireless industry and huge demand for high data rate applications. This led to the changes in the radio communication standards at a faster rate. Hardware implementation of radio communication systems makes the hardware radio obsolete, as it has to be redesigned with the changing radio standards. Software Defined Radio (SDR) overcomes this problem by implementing all or most of the radio functionality in software.

A radio transceiver consists of Baseband (BB) processing stage, Intermediate Frequency (IF) processing stage and Radio Frequency (RF) processing stage. In an ideal SDR, all the three stages are implemented in software by placing data converters immediately after the antenna. Due to practical limitations of data converters, a practical SDR architecture is obtained by placing data converters at IF stage. In our work, the design of reconfigurable digital IF stage of SDR receiver is presented.

At the IF stage of SDR receiver, if the wideband signal is digitized at Nyquist rate the narrow band radio channels gets oversampled due to the phenomenon of pass band sampling. The signal processing carried out at this sampling rate leads to high computa-

tional complexity and high power dissipation in the further stages of SDR receiver. Hence, a sample rate converter is employed to reduce the computational complexity and power dissipation in the design of digital IF stage.

In SDR receivers, it is required to decimate the signal sample rate by programmable integer rate or programmable fractional rate or both to support multi-standard radio communications. Integer rate SRC achieves SRC by large integer factors, while, the fractional rate SRC is required to make the signal sample rate power of two multiple of the symbol rate of radio standard into consideration. The process of decimation involves the design of anti-aliasing low pass filters to limit the signal bandwidth and reduce the aliasing effects. In this work, the design of a reconfigurable SRC filter with variable SRC factors and tunable spectral characteristics, while achieving minimum reconfiguration overhead and low hardware complexity required for software radio receivers, is presented.

We have analyzed various methods available in the literature to design a reconfigurable SRC filter in the digital front end design of a SDR receiver. It is observed that the coefficient-less CIC filters are suitable for attaining reconfigurable SRC by large integer factors with low hardware complexity and minimum reconfiguration overhead.

In this work, we have presented the design and implementation of SRC filter for software radio receivers employing area optimized coefficient-less CIC filters. It is observed that the frequency response of CIC filters exhibits gain droop in the pass band of interest and requires fractional rate SRC filter to achieve the required symbol rate.

We have analyzed the solutions available in the literature to restore gain droop in the pass band and design fractional rate SRC to propose a filter structure with improved spectral characteristics. We have implemented a discrete compensation filter based on inverse frequency response of CIC filters to restore the gain droop and a fractional rate SRC filter based on time domain Lagrange's interpolation polynomial. It is observed that the frequency response of time domain fractional rate interpolation filter is inferior in high frequency region. Since this problem can be addressed by employing a Farrow structure based joint compensation and interpolation filter, in which the required frequency response is approximated with a second order frequency domain polynomial, we have implemented the same for verifying the results.

In this work, we have proposed a modified joint compensation and interpolation filter with the frequency response equal to the inverse frequency response of the CIC filters. The frequency response of the proposed filter is verified by simulating using MATLAB. It is observed that this proposed filter achieves low peak gain error when compared to the filter designed using existing techniques. The proposed filter is modelled using VHDL and the functionality is verified by implementing on Kintex-7 FPGA. It is observed that this proposed modified joint compensation and interpolation filter and the existing compensation and interpolation filters have to be redesigned with the change in radio communications standard, which involves high reconfiguration overhead.

We have proposed a SVD-VDF based joint compensation and interpolation filter to design reconfigurable SRC filter with less reconfiguration overhead for multi-standard SDR receivers. The frequency response of the proposed filter and the filters considered for comparison are simulated using FDATOOL in MATLAB. It is observed that the proposed filter achieves improved spectral characteristics. It is also observed that the proposed SVD-VDF filter can be tuned to adapt to the spectral characteristics of any existing or forthcoming radio communications standard with minimum reconfiguration overhead. The proposed SVD based reconfigurable SRC filter and the filters considered for comparison are implemented on Kintex-7 FPGA and their hardware complexity is estimated. It is observed that the proposed reconfigurable SRC filter requires more hardware compared to the SRC filter designed using joint compensation and interpolation filter architecture available in the literature.

The hardware complexity of this proposed SVD-VDF is reduced by employing distributed arithmetic based FIR filter structure. We have designed SVD-VDF by employing Systolic RAM-LUT and Shared RAM-LUT based DA-FIR architectures available in the literature. These filters are modelled using VHDL and implemented on Kintex-7 FPGA. It is observed that the Systolic RAM-LUT architecture require more hardware compared to the Shared RAM-LUT architecture.

The hardware complexity of SVD-VDF employing Shared RAM-LUT architecture can be further reduced by proposing ROM-LUT based DA-FIR filter architectures. In this work, we propose the design of SVD-VDF filters by employing ROM-MUX and Systolic ROM based DA-FIR filter architectures. The hardware complexity and latency

of these proposed ROM-based DA-FIR architectures are computed and compared with the Shared RAM-LUT DA-FIR architecture. It is observed that the proposed ROM-MUX and systolic ROM-LUT based DA-FIR architectures achieves reduction in area by 18% and 70%, respectively. It is also observed that the latency of the proposed architectures is improved by two clock cycles compared to the Shared RAM-LUT DA FIR architecture available in the literature.

The proposed design of SVD-VDF employing ROM-MUX and Systolic ROM-LUT based DA-FIR architectures is modelled using VHDL and synthesized using Synopsis Design Compiler with TSMC CMOS 90nm library. From the comparison of ASIC implementation results, it is concluded that the hardware complexity of the proposed ROM-MUX architecture and Systolic ROM architecture reduces by 6% and 50%, respectively and achieves energy efficiency by more than 17% and 30%, respectively, compared to the Shared RAM-LUT DA-FIR architecture available in the literature.

## 7.2 Future Scope

This section provides the future scope of the research work presented in this thesis.

1. In this thesis, SVD-VDF based reconfigurable SRC filter designed using fixed coefficient sub-filters and multi-dimensional polynomials is proposed. In addition, Systolic ROM DA-FIR filter architecture is also developed for designing sub-filters in SVD-VDF to achieve reduction in hardware complexity and power dissipation. In future, the sub-filters in SVD-VDF can be designed as symmetric FIR filters such that the impulse response coefficients of the filter can be computed as  $h(N - n) = h(n)$ . This results in further reduction of hardware complexity approximately by a factor of two and hence reduction in power dissipation can also be achieved.
2. Error analysis of SVD-VDF based reconfigurable SRC filter employing Systolic ROM DA-FIR filter architecture can be performed to consider the effect of finite word-length used for representing input sequence and filter coefficients. In addition, the error computed for this architecture can be compared with the filter architecture designed using symmetric filter coefficients.

## Appendix A

### ASIC Synthesis Results

In this appendix we present synthesis report of Systolic ROM based DA-FIR filter architecture with  $M = 4, R = 1$  when synthesized using Synopsys Design Compiler.

## A.1 Area Report

```
*****
Report : area
Design : finaltunablecomp_1
Version: L-2016.03
Date   : Sat Apr 20 10:49:27 2019
*****
```

Information: Updating design information... (UID-85)  
Warning: Design 'finaltunablecomp\_1' contains 3 high-fanout nets. A fanout number  
of 1000 will be used for delay calculations involving these nets. (TIM-134)  
Library(s) Used:

saed90nm\_typ (File:  
/home/Synopsis1/synopsys/design\_vision/Agarwal20\_4\_19/saed90nm\_typ.db)

Number of ports:	156574
Number of nets:	514209
Number of cells:	341621
Number of combinational cells:	296640
Number of sequential cells:	42073
Number of macros/black boxes:	0
Number of buf/inv:	18785
Number of references:	12
Combinational area:	2408157.496842
Buf/Inv area:	103881.053941
Noncombinational area:	850902.460449
Macro/Black Box area:	0.000000
Net Interconnect area:	undefined (No wire load specified)
Total cell area:	3362941.011232
Total area:	undefined
1	

## A.2 Power Report

```

design_vision> report_power
Loading db file
'/home/Synopsis1/synopsys/design_vision/Agarwal20_4_19/saed90nm_typ.db'
Information: Propagating switching activity (low effort zero delay simulation).
(PWR-6)
Warning: Design has unannotated primary inputs. (PWR-414)
Warning: Design has unannotated sequential cell outputs. (PWR-415)

*****
Report : power
    -analysis_effort low
Design : finaltunablecomp_1
Version: L-2016.03
Date   : Sat Apr 20 10:52:37 2019
*****

```

Library(s) Used:

```

        saed90nm_typ (File:
/home/Synopsis1/synopsys/design_vision/Agarwal20_4_19/saed90nm_typ.db)

```

Operating Conditions: TYPICAL Library: saed90nm\_typ  
Wire Load Model Mode: top

Global Operating Voltage = 1.2  
Power-specific unit information :  
    Voltage Units = 1V  
    Capacitance Units = 1.000000pf  
    Time Units = 1ns  
    Dynamic Power Units = 1mW     (derived from V,C,T units)  
    Leakage Power Units = 1pW

Cell Internal Power = 389.5817 mW (85%)  
Net Switching Power = 77.5717 mW (15%)  
-----  
Total Dynamic Power = 467.1534 mW (100%)  
Cell Leakage Power = 10.1387 mW

Power Group	Internal Power	Switching Power	Leakage Power	Total Power
io_pad	0.0000	0.0000	0.0000	0.0000
memory	0.0000	0.0000	0.0000	0.0000
black_box	0.0000	0.0000	0.0000	0.0000
clock_network	0.0000	0.0000	0.0000	0.0000
register	194.5096	1.9194	1.9326e+09	196.3617
sequential	0.0000	0.0000	0.0000	0.0000
combinational	195.0153	75.6464	8.2061e+09	290.8747
Total 1	389.5248 mW	77.5657 mW	1.0139e+10 pW	477.2365 mW

### A.3 Timing Report

```

design_vision> report_timing
*****
Report : timing
  -path full
  -delay max
  -max_paths 1
Design : finaltunablecomp_1
Version: L-2016.03
Date   : Sat Apr 20 10:55:26 2019
*****


# A fanout number of 1000 was used for high fanout net computations.

Operating Conditions: TYPICAL  Library: saed90nm_typ
Wire Load Model Mode: top

Startpoint: svd0/c0/rtsum10_reg[4]
  (rising edge-triggered flip-flop clocked by clk)
Endpoint: svd0/c0/rtsum20_reg[31]
  (rising edge-triggered flip-flop clocked by clk)
Path Group: clk
Path Type: max

Point           Incr      Path
-----
clock clk (rise edge)      0.00      0.00
clock network delay (ideal) 0.00      0.00
svd0/c0/rtsum10_reg[4]/CLK (DFFX1) 0.00 #
svd0/c0/rtsum10_reg[4]/Q (DFFX1) 0.20      0.20 r
svd0/c0/add_650/A[4] (dafirleg_DW01_add_18) 0.00      0.20 r
svd0/c0/add_650/U119/QN (NAND2X0) 0.05      0.24 f
svd0/c0/add_650/U93/QN (INVX0) 0.04      0.28 r
svd0/c0/add_650/U80/QN (NAND2X0) 0.04      0.32 f
svd0/c0/add_650/U79/QN (NAND2X0) 0.04      0.36 r
svd0/c0/add_650/U43/QN (NAND2X0) 0.04      0.40 f
svd0/c0/add_650/U42/QN (NAND2X0) 0.04      0.43 r
svd0/c0/add_650/U41/QN (NAND2X0) 0.04      0.47 f
svd0/c0/add_650/U40/QN (NAND2X0) 0.06      0.54 r
svd0/c0/add_650/U6/Q (AND2X1) 0.07      0.61 r
svd0/c0/add_650/U5/Q (OR2X1) 0.07      0.68 r
svd0/c0/add_650/U84/QN (NAND2X0) 0.03      0.71 f
svd0/c0/add_650/U83/QN (NAND2X0) 0.04      0.75 r
svd0/c0/add_650/U82/QN (NAND2X0) 0.04      0.79 f
svd0/c0/add_650/U81/QN (NAND2X0) 0.04      0.83 r
svd0/c0/add_650/U73/QN (NAND2X0) 0.04      0.86 f
svd0/c0/add_650/U72/QN (NAND2X0) 0.04      0.90 r
svd0/c0/add_650/U71/QN (NAND2X0) 0.04      0.94 f
svd0/c0/add_650/U70/QN (NAND2X0) 0.05      0.99 r
svd0/c0/add_650/U45/QN (NAND2X0) 0.04      1.03 f
svd0/c0/add_650/U44/QN (NAND2X0) 0.05      1.09 r
svd0/c0/add_650/U47/QN (NAND2X0) 0.04      1.13 f
svd0/c0/add_650/U46/QN (NAND2X0) 0.05      1.18 r
svd0/c0/add_650/U49/QN (NAND2X0) 0.04      1.22 f
svd0/c0/add_650/U48/QN (NAND2X0) 0.05      1.27 r
svd0/c0/add_650/U51/QN (NAND2X0) 0.04      1.31 f
svd0/c0/add_650/U50/QN (NAND2X0) 0.05      1.36 r
svd0/c0/add_650/U53/QN (NAND2X0) 0.04      1.41 f
svd0/c0/add_650/U52/QN (NAND2X0) 0.05      1.46 r

```

svd0/c0/add_650/U55/QN (NAND2X0)	0.04	1.50 f
svd0/c0/add_650/U54/QN (NAND2X0)	0.05	1.55 r
svd0/c0/add_650/U57/QN (NAND2X0)	0.04	1.59 f
svd0/c0/add_650/U56/QN (NAND2X0)	0.05	1.64 r
svd0/c0/add_650/U59/QN (NAND2X0)	0.04	1.69 f
svd0/c0/add_650/U58/QN (NAND2X0)	0.05	1.74 r
svd0/c0/add_650/U61/QN (NAND2X0)	0.04	1.78 f
svd0/c0/add_650/U60/QN (NAND2X0)	0.05	1.83 r
svd0/c0/add_650/U63/QN (NAND2X0)	0.04	1.87 f
svd0/c0/add_650/U62/QN (NAND2X0)	0.05	1.92 r
svd0/c0/add_650/U65/QN (NAND2X0)	0.04	1.96 f
svd0/c0/add_650/U64/QN (NAND2X0)	0.05	2.02 r
svd0/c0/add_650/U75/QN (NAND2X0)	0.04	2.06 f
svd0/c0/add_650/U74/QN (NAND2X0)	0.05	2.11 r
svd0/c0/add_650/U160/QN (NAND2X0)	0.04	2.15 f
svd0/c0/add_650/U158/QN (NAND2X0)	0.05	2.20 r
svd0/c0/add_650/U164/QN (NAND2X0)	0.04	2.24 f
svd0/c0/add_650/U162/QN (NAND2X0)	0.05	2.29 r
svd0/c0/add_650/U166/QN (NAND2X0)	0.04	2.34 f
svd0/c0/add_650/U171/QN (NAND2X0)	0.05	2.38 r
svd0/c0/add_650/U169/Q (XOR2X1)	0.12	2.41 r
svd0/c0/add_650/SUM[31] (dafirleg_DW01_add_18)	0.00	2.41 r
svd0/c0/rtsum20_reg[31]/D (DFFX1)	0.00	2.41 r
data arrival time		2.41
-----		
clock clk (rise edge)	3.00	3.00
clock network delay (ideal)	0.00	3.00
svd0/c0/rtsum20_reg[31]/CLK (DFFX1)	0.00	3.00 r
library setup time	-0.07	2.93
data required time		2.93
-----		
data required time		2.93
data arrival time		-2.41
-----		
slack (MET)		0.52

# Publications

---

## List of International Journals:

---

1. Agarwal Ashok, and B. Lakshmi, "FPGA Implementation of Optimized CIC Filter for Sample Rate Conversion in Software Radio Receiver," *International Journal on Recent Trends in Engineering and Technology* 9, no. 1 (2013): 60. **ISI-Indexed**
2. Agarwal Ashok, and Lakshmi Bopanna, "Design of reconfigurable pipelined Architecture For SRC Filter for software radio receiver," *International Journal of Electronics and Communication Engineering* 1, no. 4(2015): 23-30 **Peer reviewed**
3. Agarwal Ashok, and Lakshmi Bopanna, "SVD based reconfigurable SRC filter for multi-standard radio receivers," *IET Circuits, Devices & Systems* 10, no. 5 (2016): 375-382. **SCI-Indexed**
4. Agarwal Ashok, and Lakshmi Bopanna, "Low Latency Area-Efficient Distributed Arithmetic Based Multi-Rate Filter Architecture for SDR Receivers," *Journal of Circuits, Systems and Computers* 27, no. 08 (2018): 1850133. **SCI-Indexed**

---

## List of International Conferences:

---

1. Agarwal Ashok, and Bopanna Lakshmi, "FPGA implementation of digital down converter using CORDIC algorithm," in *International Conference on Communication and Electronics System Design*, vol. 8760, p. 87601K. International Society for Optics and Photonics, 2013.
2. Agarwal Ashok, Lakshmi Bopanna, and Ravi Kishore Kodali, "A factorization method for FPGA implementation of sample rate converter for a multi-standard

radio communications," in *TENCON Spring Conference, 2013 IEEE*, pp. 530-534, IEEE 2013.

3. Lakshmi B., and Ashok Agarwal, "An Optimized Sample Rate Converter for a software radio receiver on FPGA," in *Atlantis Press*, 2013.
4. Agarwal Ashok, Lakshmi Boppana, and Ravi Kishore Kodali, "Lagrange's polynomial based farrow filter implementation for SDR," In *Region 10 Symposium, IEEE* pp. 269-274. IEEE 2014.
5. Agarwal Ashok, Lakshmi Boppana, and Ravi Kishore Kodali, "A fractional sample rate conversion filter for a software radio receiver on FPGA," in *TENCON 2014 IEEE Region 10 Conference*, pp. 1-6. IEEE 2014.

## Bibliography

- [1] K.-m. Tsui, “Efficient design and realization of digital IFs and time-interleaved analog-to-digital converters for software radio receivers,” 2008.
- [2] J. Mitola, “Software radios: Survey, critical evaluation and future directions,” *IEEE Aerospace and Electronic Systems Magazine*, vol. 8, no. 4, pp. 25–36, 1993.
- [3] Joseph Mitola, “The software radio architecture,” *IEEE Communications magazine*, vol. 33, no. 5, pp. 26–38, 1995.
- [4] R. H. Walden, “Analog-to-digital converter survey and analysis,” *IEEE Journal on selected areas in communications*, vol. 17, no. 4, pp. 539–550, 1999.
- [5] P. Burns, *Software defined radio for 3G*. Artech house, 2006.
- [6] T. Hentschel, M. Henker, and G. Fettweis, “The digital front-end of software radio terminals,” *IEEE Personal communications*, vol. 6, no. 4, pp. 40–46, 1999.
- [7] T. Hentschel, “Channelization for software defined base-stations,” in *Annales des Telecommunications*, vol. 57, no. 5-6. Springer, 2002, p. 386.
- [8] C. Y. Fung and S. Chan, “A multistage filterbank-based channelizer and its multiplier-less realization,” in *Circuits and Systems, 2002. ISCAS 2002. IEEE International Symposium on*, vol. 3. IEEE, 2002, pp. III–III.
- [9] W. A. Abu-Al-Saud and G. L. Stuber, “Efficient wideband channelizer for software radio systems using modulated PR filterbanks,” *IEEE transactions on signal processing*, vol. 52, no. 10, pp. 2807–2820, 2004.
- [10] J. Lillington, “TPFT–tunable pipelined frequency transform,” *RF Engines Ltd, Technical Report. <http://www.rfel.com/whipapdat.asp>*, 2002.

- [11] A. P. Vinod, E.-K. Lai, A. B. Premkumar, and C. T. Lau, “A reconfigurable multi-standard channelizer using QMF trees for software radio receivers,” in *Personal, Indoor and Mobile Radio Communications, 2003. PIMRC 2003. 14th IEEE Proceedings on*, vol. 1. IEEE, 2003, pp. 119–123.
- [12] J. Lillington, “Comparison of wideband channelisation architectures,” in *International Signal Processing Conference (ISPC), Dallas*, 2003.
- [13] Y. Lim, “Frequency-response masking approach for the synthesis of sharp linear phase digital filters,” *IEEE transactions on circuits and systems*, vol. 33, no. 4, pp. 357–364, 1986.
- [14] R. Mahesh, A. P. Vinod, C. Moy, and J. Palicot, “A low complexity reconfigurable filter bank architecture for spectrum sensing in cognitive radios,” in *Cognitive Radio Oriented Wireless Networks and Communications, 2008. CrownCom 2008. 3rd International Conference on*. IEEE, 2008, pp. 1–6.
- [15] R. Mahesh and A. P. Vinod, “Reconfigurable frequency response masking filters for software radio channelization,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 55, no. 3, pp. 274–278, 2008.
- [16] Mahesh R and Vinod, Achutavarrier Prasad, “Coefficient decimation approach for realizing reconfigurable finite impulse response filters,” in *Circuits and Systems, 2008. ISCAS 2008. IEEE International Symposium on*. IEEE, 2008, pp. 81–84.
- [17] R. Mahesh, A. P. Vinod, E. M. Lai, and A. Omondi, “Filter bank channelizers for multi-standard software defined radio receivers,” *Journal of signal processing systems*, vol. 62, no. 2, pp. 157–171, 2011.
- [18] J. E. Volder, “The CORDIC trigonometric computing technique,” *IRE Transactions on electronic computers*, no. 3, pp. 330–334, 1959.
- [19] J. S. Walther, “A unified algorithm for elementary functions,” in *Proceedings of the May 18-20, 1971, spring joint computer conference*. ACM, 1971, pp. 379–385.
- [20] B. Lakshmi and A. Dhar, “CORDIC architectures: a survey,” *VLSI design*, vol. 2010, p. 2, 2010.

- [21] T. Hentschel and G. Fettweis, "Sample rate conversion for software radio," *IEEE Communications magazine*, vol. 38, no. 8, pp. 142–150, 2000.
- [22] S. Chan and K. Yeung, "On the design and multiplier-less realization of digital IF for software radio receivers with prescribed output accuracy," in *Digital Signal Processing, 2002. DSP 2002. 2002 14th International Conference on*, vol. 1. IEEE, 2002, pp. 277–280.
- [23] E. M. Hofstetter, "A new technique for the design of non-recursive digital filters," MASSACHUSETTS INST OF TECH LEXINGTON LINCOLN LAB, Tech. Rep., 1970.
- [24] L. R. Rabiner, "The design of finite impulse response digital filters using linear programming techniques," *Bell System Technical Journal*, vol. 51, no. 6, pp. 1177–1198, 1972.
- [25] T. Parks and J. McClellan, "Chebyshev approximation for nonrecursive digital filters with linear phase," *IEEE Transactions on Circuit Theory*, vol. 19, no. 2, pp. 189–194, 1972.
- [26] J. F. Kaiser, "Nonrecursive digital filter design using the i\_0-sinh window function," in *Proc. 1974 IEEE International Symposium on Circuits & Systems, San Francisco DA, April*, 1974, pp. 20–23.
- [27] R. W. Schafer and L. R. Rabiner, "A digital signal processing approach to interpolation," *Proceedings of the IEEE*, vol. 61, no. 6, pp. 692–702, 1973.
- [28] R. Crochiere and L. Rabiner, "Optimum FIR digital filter implementations for decimation, interpolation, and narrow-band filtering," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 23, no. 5, pp. 444–456, 1975.
- [29] T. Claasen and W. Mecklenbräuker, "On the transposition of linear time-varying discrete-time networks and its application to multirate digital systems," *Philips journal of research*, vol. 33, no. 1, pp. 78–102, 1978.
- [30] M. Bellanger, G. Bonnerot, and M. Coudreuse, "Digital filtering by polyphase network: Application to sample-rate alteration and filter banks," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 2, pp. 109–114, 1976.

- [31] J. McClellan, T. Parks, and L. Rabiner, “A computer program for designing optimum FIR linear phase digital filters,” *IEEE Transactions on Audio and Electroacoustics*, vol. 21, no. 6, pp. 506–526, 1973.
- [32] D. Rorabacher, “Efficient FIR filter design for sample rate reduction or interpolation,” in *Proc. 1975 IEEE Int. Symp. on Circuits and Systems*, 1975, pp. 396–399.
- [33] M. Bellanger, J. Daguet, and G. Lepagnol, “Interpolation, extrapolation, and reduction of computation speed in digital filters,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 22, no. 4, pp. 231–235, 1974.
- [34] M. Bellanger, “Computation rate and storage estimation in multirate digital filtering with half-band filters,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 25, no. 4, pp. 344–346, 1977.
- [35] R. Shively, “On multistage finite impulse response (fir) filters with decimation,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 23, no. 4, pp. 353–357, 1975.
- [36] R. Crochiere and L. Rabiner, “A program for multistage decimation, interpolation and narrow-band filtering,” *Programs for Digital Signal Processing*, C.J. Weinstein, editor, IEEE Press, New York, no. 0, p. 0, 1979.
- [37] R. E. Crochiere and L. R. Rabiner, “Interpolation and decimation of digital signals: A tutorial review,” *Proceedings of the IEEE*, vol. 69, no. 3, pp. 300–331, 1981.
- [38] A. Peled and B. Liu, “A new hardware realization of digital filters,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 22, no. 6, pp. 456–462, 1974.
- [39] D. Goodman and M. Carey, “Nine digital filters for decimation and interpolation,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 25, no. 2, pp. 121–126, 1977.
- [40] E. Hogenauer, “An economical class of digital filters for decimation and interpolation,” *IEEE transactions on acoustics, speech, and signal processing*, vol. 29, no. 2, pp. 155–162, 1981.

- [41] T. Ramstad, “Digital methods for conversion between arbitrary sampling frequencies,” *IEEE transactions on acoustics, speech, and signal processing*, vol. 32, no. 3, pp. 577–591, 1984.
- [42] C. W. Farrow, “A continuously variable digital delay element,” in *Circuits and Systems, 1988., IEEE International Symposium on.* IEEE, 1988, pp. 2641–2645.
- [43] G. Liu and C. Wei, “Programmable fractional sample delay filter with Lagrange interpolation,” *Electronics Letters*, vol. 26, no. 19, pp. 1608–1610, 1990.
- [44] K. Pun, Y. Wu, S. Chan, and K. Ho, “An efficient design of fractional-delay digital FIR filters using the Farrow structure,” in *IEEE Workshop on Statistical Signal Processing Proceedings.* IEEE., 2001.
- [45] F. M. Gardner, “Interpolation in digital modems-Part I:Fundamentals,” *IEEE Transactions on communications*, vol. 41, no. 3, pp. 501–507, 1993.
- [46] L. Erup, F. M. Gardner, and R. A. Harris, “Interpolation in digital modems-Part II: Implementation and performance,” *IEEE Transactions on communications*, vol. 41, no. 6, pp. 998–1008, 1993.
- [47] V. Valimaki, “A new filter implementation strategy for Lagrange interpolation,” in *Circuits and Systems, 1995. ISCAS’95., 1995 IEEE International Symposium on,* vol. 1. IEEE, 1995, pp. 361–364.
- [48] H. Johansson and P. Lowenborg, “On the design of adjustable fractional delay FIR filters,” *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 50, no. 4, pp. 164–169, 2003.
- [49] W. A. Abu-Al-Saud and G. L. Stuber, “Modified CIC filter for sample rate conversion in software radio systems,” *IEEE Signal Processing Letters*, vol. 10, no. 5, pp. 152–154, 2003.
- [50] C. Candan, “An efficient filtering structure for Lagrange interpolation,” *IEEE Signal Process. Lett.*, vol. 14, no. 1, pp. 17–19, 2007.
- [51] R. Zarour and M. M. Fahmy, “A design technique for variable digital filters,” *IEEE Transactions on Circuits and Systems*, vol. 36, no. 11, pp. 1473–1478, 1989.

- [52] P. E. Gill and W. Murray, "Algorithms for the solution of the nonlinear least-squares problem," *SIAM Journal on Numerical Analysis*, vol. 15, no. 5, pp. 977–992, 1978.
- [53] T.-B. Deng and T. Soma, "Variable digital filter design using the outer product expansion," *IEE Proceedings-Vision, Image and Signal Processing*, vol. 141, no. 2, pp. 123–128, 1994.
- [54] T.-B. Deng, "Weighted least-squares design of variable 1-D FIR filters with arbitrary magnitude responses," in *Circuits and Systems, 2000. IEEE APCCAS 2000. The 2000 IEEE Asia-Pacific Conference on.* IEEE, 2000, pp. 288–291.
- [55] T.-B. Deng and W.-S. Lu, "Weighted least-squares method for designing variable fractional delay 2-D FIR digital filters," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 47, no. 2, pp. 114–124, 2000.
- [56] C.-C. Tseng, "Design of 1-D and 2-D variable fractional delay allpass filters using weighted least-squares method," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 49, no. 10, pp. 1413–1422, 2002.
- [57] T.-B. Deng, "Discretization-free design of variable fractional-delay FIR digital filters," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 48, no. 6, pp. 637–644, 2001.
- [58] Deng, Tian-Bo, "Closed-form design and efficient implementation of variable digital filters with simultaneously tunable magnitude and fractional delay," *IEEE Transactions on Signal Processing*, vol. 52, no. 6, pp. 1668–1681, 2004.
- [59] T.-B. Deng and Y. Nakagawa, "SVD-based design and new structures for variable fractional-delay digital filters," *IEEE Transactions on Signal Processing*, vol. 52, no. 9, pp. 2513–2527, 2004.
- [60] T.-B. Deng, "Design of complex-coefficient variable digital filters using successive vector-array decomposition," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 52, no. 5, pp. 932–942, 2005.
- [61] Deng, Tian-Bo, "Design of arbitrary-phase variable digital filters using SVD-based vector-array decomposition," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 52, no. 1, pp. 148–167, 2005.

- [62] S. Chu and C. Burrus, "Multirate filter designs using comb filters," *IEEE Transactions on Circuits and Systems*, vol. 31, no. 11, pp. 913–924, 1984.
- [63] J.-K. Liang and R. de Figueiredo, "A new class of nonlinear phase FIR digital filters and its application to efficient design of multirate digital filters," *IEEE transactions on circuits and systems*, vol. 32, no. 9, pp. 944–948, 1985.
- [64] F. J. Harris, "Multirate FIR filters for interpolating and desampling," in *Handbook of digital signal processing*. Elsevier, 1987, pp. 173–287.
- [65] A. Y. Kwentus, Z. Jiang, and A. N. Willson, "Application of filter sharpening to cascaded integrator-comb decimation filters," *IEEE Transactions on Signal Processing*, vol. 45, no. 2, pp. 457–467, 1997.
- [66] T. Saramaki and T. Ritoniemi, "A modified comb filter structure for decimation," in *Circuits and Systems, 1997. ISCAS'97., Proceedings of 1997 IEEE International Symposium on*, vol. 4. IEEE, 1997, pp. 2353–2356.
- [67] F. Daneshgaran and M. Laddomada, "A novel class of decimation filters for  $\sigma\delta$  a/d converters," *Wireless Communications and Mobile Computing*, vol. 2, no. 8, pp. 867–882, 2002.
- [68] G. Jovanovic-Dolecek and S. Mitral, "Efficient sharpening of CIC decimation filter," in *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP'03). 2003 IEEE International Conference on*, vol. 6. IEEE, 2003, pp. VI–385.
- [69] G. Jovanovic-Dolecek and S. K. Mitra, "A new two-stage sharpened comb decimator," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 52, no. 7, pp. 1414–1420, 2005.
- [70] B. P. Brandt and B. A. Wooley, "A low-power, area-efficient digital filter for decimation and interpolation," *IEEE Journal of Solid-State Circuits*, vol. 29, no. 6, pp. 679–687, 1994.
- [71] H. J. Oh, S. Kim, G. Choi, and Y. H. Lee, "On the use of interpolated second-order polynomials for efficient filter design in programmable downconversion," *IEEE Journal on selected areas in communications*, vol. 17, no. 4, pp. 551–560, 1999.

- [72] L. Rabiner, "Linear program design of finite impulse response (fir) digital filters," *IEEE Transactions on Audio and Electroacoustics*, vol. 20, no. 4, pp. 280–288, 1972.
- [73] Y. Lim and S. Parker, "FIR filter design over a discrete powers-of-two coefficient space," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 31, no. 3, pp. 583–591, 1983.
- [74] K. Yeung and S.-C. Chan, "The design and multiplier-less realization of software radio receivers with reduced system delay," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 51, no. 12, pp. 2444–2459, 2004.
- [75] S. Kim, W.-C. Lee, S. Ahn, and S. Choi, "Design of CIC roll-off compensation filter in a W-CDMA digital IF receiver," *Digital Signal Processing*, vol. 16, no. 6, pp. 846–854, 2006.
- [76] G. J. Dolecek and S. K. Mitra, "A new two-stage CIC-based decimation filter," in *Image and Signal Processing and Analysis, 2007. ISPA 2007. 5th International Symposium on.* IEEE, 2007, pp. 218–223.
- [77] Dolecek, G Jovanovic and Mitra, Sanjit K, "Simple method for compensation of CIC decimation filter," *Electronics Letters*, vol. 44, no. 19, pp. 1162–1163, 2008.
- [78] G. J. Dolecek, "Simple wideband CIC compensator," *Electronics Letters*, vol. 45, no. 24, pp. 1270–1272, 2009.
- [79] G. J. Dolecek *et al.*, "Design of wideband CIC compensator filter for a digital IF receiver," *Digital Signal Processing*, vol. 19, no. 5, pp. 827–837, 2009.
- [80] G. Molnar and M. Vucic, "Closed-form design of CIC compensators based on maximally flat error criterion," *Rn*, vol. 2, p. 2, 2011.
- [81] T. Instruments, "Gc4016 multi-standard quad DDC chip data sheet," *data manual Revision*, vol. 1, no. 70-90, p. 2009, 2001.
- [82] A. Fernandez-Vazquez and G. Jovanovic-Dolecek, "Maximally flat CIC compensation filter: Design and multiplierless implementation." *IEEE Trans. on Circuits and Systems*, vol. 59, no. 2, pp. 113–117, 2012.

- [83] M. G. Pecotic, G. Molnar, and M. Vucic, “Design of CIC compensators with SPT coefficients based on interval analysis,” in *MIPRO, 2012 Proceedings of the 35th International Convention*. IEEE, 2012, pp. 123–128.
- [84] A. Franck and K. Brandenburg, “An overall optimization method for arbitrary sample rate converters based on integer rate SRC and Lagrange interpolation,” in *Applications of Signal Processing to Audio and Acoustics, 2009. WASPAA’09. IEEE Workshop on*. IEEE, 2009, pp. 301–304.
- [85] F. Sheikh and S. Masud, “Sample rate conversion filter design for multi-standard software radios,” *Digital Signal Processing*, vol. 20, no. 1, pp. 3–12, 2010.
- [86] X.-G. Xia, “Fractional delay filter design when sampling rate higher than nyquist rate,” *Electronics Letters*, vol. 33, no. 3, pp. 199–201, 1997.
- [87] J. Vesma, “A frequency-domain approach to polynomial-based interpolation and the farrow structure,” *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 47, no. 3, pp. 206–209, 2000.
- [88] R. D. Carsello, R. Meidan, S. Allpress, F. O’Brien, J. A. Tarallo, N. Ziesse, A. Arunachalam, J. M. Costa, E. Berruto, R. C. Kirby *et al.*, “IMT-2000 standards: Radio aspects,” *IEEE Personal Communications*, vol. 4, no. 4, pp. 30–40, 1997.
- [89] A. G. Dempster and M. D. Macleod, “Use of minimum-adder multiplier blocks in FIR digital filters,” *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 42, no. 9, pp. 569–577, 1995.
- [90] Y. C. Lim, R. Yang, D. Li, and J. Song, “Signed power-of-two term allocation scheme for the design of digital filters,” *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 46, no. 5, pp. 577–584, 1999.
- [91] S. A. White, “Applications of distributed arithmetic to digital signal processing: A tutorial review,” *IEEE Assp Magazine*, vol. 6, no. 3, pp. 4–19, 1989.
- [92] P. K. Meher, “Hardware-efficient systolization of DA-based calculation of finite digital convolution,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 53, no. 8, pp. 707–711, 2006.

- [93] E. Ozalevli, W. Huang, P. E. Hasler, and D. V. Anderson, “A reconfigurable mixed-signal VLSI implementation of distributed arithmetic used for finite-impulse response filtering,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 55, no. 2, pp. 510–521, 2008.
- [94] P. K. Meher, S. Chandrasekaran, and A. Amira, “FPGA realization of FIR filters by efficient and flexible systolization using distributed arithmetic,” *IEEE transactions on signal processing*, vol. 56, no. 7, pp. 3009–3017, 2008.
- [95] S. Y. Park and P. K. Meher, “Efficient FPGA and ASIC realizations of a DA-based reconfigurable FIR digital filter,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 61, no. 7, pp. 511–515, 2014.
- [96] B. K. Mohanty, P. K. Meher, S. K. Singhal, and M. Swamy, “A high-performance VLSI architecture for reconfigurable FIR using distributed arithmetic,” *Integration, the VLSI Journal*, vol. 54, pp. 37–46, 2016.