Low power reconfigurable digital filter banks for multi-standard wireless communications receivers

Kavallur Pisharath Gopi Smitha

2010

Kavallur, P. G.S. (2010). Low power reconfigurable digital filter banks for multi-standard wireless communications receivers. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/47495>

<https://doi.org/10.32657/10356/47495>

Nanyang Technological University

Downloaded on 26 Apr 2023 12:32:59 SGT

4x
32

LOW POWER RECONFIGURABLE DIGITAL FILTER BANKS FOR MULTI-STANDARD WIRELESS COMMUNICATION RECEIVERS

KAVALLUR PISHARATH GOPI SMITHA

School of Computer Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Doctor of Philosophy

2010

Acknowledgements

I would like to offer my sincerest gratitude to my supervisor, Dr. Vinod Achutavarrier Prasad, for his invaluable guidance and encouragement throughout this research work. His expertise, enthusiasm in research and positive way of thinking has been of great value for me. The quality of the work and its publication achievement if any, owe much to his inputs, constructive criticisms and patient guidance. I am indebted to him for his personal advises and moral support which enabled me to complete my work successfully.

I am grateful to Dr. Hossam A. H. Fahmy, for helping me to find a direction during the initial years of PhD work. I would like to thank all my friends for their support throughout the period of this work. In particular, I wish to thank Dr. R. Mahesh, for his friendship and help in the past four years. I would also like to thank the Centre for High Performance Embedded Systems for offering all the facilities and School of Computer Engineering for the financial support provided in carrying out this research.

I owe my deepest gratitude to my husband, Dr. Sreekumar Pisharath, for his constant support. Thanks to dear daughter, Shreya for helping to relax and refresh myself with her little pranks. Last but not the least, I thank and dedicate this work to my loving parents, for the way they brought me up in life.

Contents

Acknowledgementsi

Table of Contentsii

List of Figuresvi

List of Tablesx

List of Abbreviationsxii

Abstractxv

Chapter 11

Introduction1

1.1 Motivation4

1.2 Objectives and Contributions5

1.3 Outline8

Chapter 29

Software Defined Radios: A Literature Survey .. 9

2.1 Definition and Objectives SDR9

2.2 SDR Receiver Architecture10

2.3 SDR Receiver Functionalities13

2.3.1 Analog-to-Digital Conversion14

2.3.2 Digital Front End15

2.3.2.1 Digital Down Conversion16

2.3.2.2 Channelization17

2.3.2.3 Sample Rate Conversion17

2.4 Channelization for SDR Receivers18

2.5 Channelization Architectures20

2.5.1 The Per-Channel Approach20

2.5.2 The Filter Bank Approach22

2.5.3	The Goertzel Filter Bank	25
2.5.4	The Pipelined Frequency Transform.....	25
2.5.5	The Tree structured Quadrature Mirror Filter Bank	27
2.5.6	The Modified Perfect Reconstruction Filter Bank	27
2.5.7	The Multi-Stage Filter Bank	28
2.5.8	The Cosine Modulated Filter Bank	29
2.6	Qualitative Comparison of Digital Filter Banks	29
2.7	Summary.....	31
Chapter 3		33
Low Power and Reconfigurable Implementation Approaches for Channel Filters		33
3.1	Channel Filters for Wideband Receivers	33
3.2	Computational Complexity Analysis of Channel Filters	37
3.3	Filter Reconfigurability	47
3.4	Summary.....	49
Chapter 4		50
Low Power FIR Channel Filter Realization Using Binary Common Subexpression Elimination		50
4.1	Binary Common Subexpression Elimination Algorithm	50
4.1.1	Binary Horizontal Subexpression Elimination (BHSE)	59
4.1.2	Binary Horizontal Super Subexpression Elimination (BSSE)	60
4.2	Extension of BCSE-4 to High-Level Synthesis	62
4.3	Design Examples	63
4.4	Synthesis and Implementation Results	72
4.5	Summary	74
Chapter 5		75
Reconfigurable Low Complexity Filters and Filter Banks based on Multi-band Filtering and Masking Technique		75
5.1	Proposed Multi-band Filter and Masking Method	76

5.2	Experimental Results	79
5.2.1	Design Examples	79
5.2.2	Functionality Test	82
5.2.3	Implementation Results	84
5.3	Practical Design Considerations of the Proposed Multi-band Filter and Masking Method	86
5.4	Multi-stage Frequency Response Masking (MSFRM) Channel Filter	92
5.4.1	Review of Frequency Response Masking	92
5.4.2	Design Steps of Proposed Multi-stage Channel Filter	96
5.5	An Example of MSFRM Based Channel Filter Architecture	98
5.5.1	Comparison of the Proposed MSFRM Method with Single-Stage FRM	102
5.5.2	Implementation Results	104
5.5.3	Design Examples	106
5.5.4	Functionality Test	110
5.6	MSFRM Based Filter Bank Architecture for SDR	112
5.6.1	Quantitative Comparison of MSFRM based Channelizer with other Methods	115
5.7	Summary	116
Chapter 6		118
A Channel Filter Architecture for Fractional Bandwidth Channel Extraction using Decimation, Interpolation and Masking.....		118
6.1	Proposed Decimation, Interpolation and Masking (DIM) based Channel Filter	119
6.1.1	Design Flow of the Proposed Channel Filter	119
6.1.2	Filter Level Reconfigurability	122
6.1.3	Architecture Level Reconfigurability.....	125
6.2	Implementation Results	126
6.3	Design Example.....	129
6.4	Functionality Test.....	132
6.5	Summary	134

Chapter 7136

A Reconfigurable High-Speed Residue Number System Based FIR Filter136

7.1 A Review on RNS-FIR Filters136

7.2 Proposed Reconfigurable RNS-FIR Filter140

7.2.1 Proposed Reconfigurable Channel Filter Architecture140

7.2.2 Product Encoder Synthesis Result143

7.3 Design Example and Synthesis Result144

7.4 Summary147

Chapter 8148

Conclusions and Future Work148

8.1 Conclusions148

8.2 Future Work154

8.2.1 One-shot Design of Reconfigurable Channel Filters using FRM technique.....154

8.2.2 Channelizers based on Fast Filter Banks.....154

8.2.3 Optimization of RNS-FIR Filters for Multiple Moduli.....155

Bibliography156

List of Publications162

List of Figures

Figure 2.1	The ideal software defined radio receiver	10
Figure 2.2	A feasible software radio receiver	12
Figure 2.3	Digital Front End	15
Figure 2.4	Bandpass ADC followed by digital down conversion	17
Figure 2.5	Principle of PC approach for channelization.....	21
Figure 2.6	DFT filter bank.....	23
Figure 2.7	Goertzel filter bank.....	25
Figure 2.8	The pipelined frequency transform architecture.....	26
Figure 2.9	Modified perfect reconstruction filter bank.....	27
Figure 2.10	Multi-stage filter bank.....	29
Figure 3.1	Direct form FIR filter structure	35
Figure 3.2(a)	Transposed Direct form FIR filter structure	36
Figure 3.2(b)	Modified TDF FIR filter structure with a common multiplier block.....	36
Figure 4.1	Filter realization using horizontal CSE	54
Figure 4.2	Filter realization using BSE	56
Figure 4.3	Filter realization using proposed BCSE-4	57
Figure 4.4(a)	Coefficient h_1 grouped using 3-bit BHCSs	60
Figure 4.4(b)	Coefficient h_1 grouped using 4-bit BHCSs	60
Figure 4.5(a)	Subexpression sharing as a high-level synthesis transformation. Conventional 2-bit CSE.....	63
Figure 4.5(b)	Subexpression sharing as a high-level synthesis transformation. Proposed 4-bit CSE (BCSE-4).....	63
Figure 4.6	Percentage reduction of LOs for 24-bit wordlength filter used in example III using NR-SCSE[40], H-CUB[52], BSE[47], CRA[51] and proposed BCSE-4 over direct method.....	69
Figure 4.7	Percentage reduction of LOs for 940-tap D-AMPS filter using NR-SCSE [40], SS [45], BSE [47], CRA [51] and proposed BCSE-4 over direct method.	71

Figure 4.8	Percentage reduction of LOs for 16-bit wordlength D-AMPS filter using NR-SCSE [40], SS [45], BSE [47], CRA [51] and proposed BCSE-4 over direct method.	72
Figure 5.1	Block diagram of proposed multi-band and masking based channel filter architecture	77
Figure 5.2	Frequency response illustration of multi-band filtering and masking based channel filter	78
Figure 5.3	Magnitude response of multi-band filter and the corresponding masking filters (TDMA/PDC).	80
Figure 5.4	Magnitude response of multi-band filter and the corresponding masking filters (CDMA/WCDMA).....	81
Figure 5.5(a)	Input spectrum	82
Figure 5.5(b)	Output spectrum of multi-band filter	82
Figure 5.5(c)	CDMA output spectrum extracted using multi-band filter and masking based architecture	83
Figure 5.5(d)	WCDMA output spectrum extracted using multi-band filter and masking based architecture	83
Figure 5.6	Simulink diagram for the CDMA/WCDMA channelizer using Lyrtech Signalmaster.	84
Figure 5.7	Frequency response of multi-band filter.	87
Figure 5.8	Magnitude response of PDC/D-AMPS/PHS channel filter.....	90
Figure 5.9	Complexity analysis of PC architecture and the proposed architecture for varying inter-channel distance (D_i).....	91
Figure 5.10	FIR filter architecture based on FRM technique	93
Figure 5.11	Frequency response illustration of FRM	93
Figure 5.12	Architecture of proposed two-stage FRM	95
Figure 5.13	Proposed MSFRM based channel filter architecture	98
Figure 5.14(a)	Modal filter architecture	101
Figure 5.14(b)	Architecture of first set masking filters (F_{MA1} , F_{MA2} , F_{MA3} , F_{MC2} , F_{MC3})	101
Figure 5.15(a)	Complementary delay architecture for F_{MC2}	102

Figure 5.15(b) Complementary delay architecture for F_{MC3}	102
Figure 5.16 SSFRM based channel filter architecture [67]	103
Figure 5.17 Implementation diagram for the 12-mode MSFRM based channel filter architecture	105
Figure 5.18(a) CDMA interpolated modal filter response and first stage masking filter response.....	107
Figure 5.18(b) CDMA interpolated complementary modal filter response and first stage complementary masking response.....	108
Figure 5.18(c) CDMA addition result of the extracted output from figures (a) and (b) (which is the CDMA channel) and masked using second stage masking filter	108
Figure 5.18(d) WCDMA interpolated modal filter response and first stage masking filter response.....	109
Figure 5.18(e) WCDMA interpolated complementary modal filter response and first stage complementary masking response.....	109
Figure 5.18(f) WCDMA addition result of the extracted output from figures (d) and (e) (which is the WCDMA channel) and masked using second stage masking filter	110
Figure 5.19(a) Input signal spectrum and CDMA spectrum extracted using proposed MSFRM based channel filter	111
Figure 5.19(b) Input signal spectrum and WCDMA spectrum extracted using proposed MSFRM based channel filter	111
Figure 5.20 Proposed MSFRM based filter bank architecture	113
Figure 5.21 SSFRM based filter bank architecture	114
Figure 6.1 Frequency response illustration of the proposed DIM based reconfigurable channel filter	120
Figure 6.2 The proposed DIM based reconfigurable channel filter for SDR	121
Figure 6.3 Architecture of modal filter, $F_a(z^{LD})$. The filter coefficients are indicated as h_0, h_1 etc and the filter is a symmetric FIR filter	123

Figure 6.4	Simulink diagram of the proposed DIM based channel filter architecture.....	127
Figure 6.5(a)	Magnitude response of the modal filter and masking filter for extracting CDMA signal.	130
Figure 6.5(b)	Magnitude response of the channel filter for extracting CDMA Signal.....	131
Figure 6.6(a)	Magnitude response of the modal filter and masking filter for extracting WCDMA signal.	131
Figure 6.6(b)	Magnitude response of the channel filter for extracting WCDMA Signal.....	132
Figure 6.7(a)	Input signal spectrum and CDMA spectrum extracted using DIM based channel filter.....	133
Figure 6.7(b)	Input signal spectrum and WCDMA spectrum extracted using DIM based channel filter.....	133
Figure 7.1	FIR filter implementation using RNS	138
Figure 7.2	$ FIR _{m_i}$ Structure	139
Figure 7.3	Reconfigurable RNS-FIR filter for $mod\ m_i$	142
Figure 7.4	High-speed and reconfigurable RNS-FIR filter	145

List of Tables

Table 2.1	Comparison of ADC technologies	15
Table 2.1	Comparison of channelization approaches	30
Table 3.1	FIR filter coefficients.....	44
Table 4.1	Binary representation of 8-tap FIR filter coefficients	53
Table 4.2	Representation of the filter coefficients for BHSSE	61
Table 4.3	Representation of the filter coefficients after BHSSE	62
Table 4.4	Comparison of LOs and LDs for benchmark FIR filters in example I.....	65
Table 4.5	Comparison of LOs and LDs of filters in example II.....	66
Table 4.6	Number of LOs for the filters in example III.....	67
Table 4.7	Number of LDs for the filters in example III.....	68
Table 4.8	Comparison of LOs for D-AMPS filters in example IV.....	70
Table 4.9	Comparison of LDs for D-AMPS filters in example IV.....	70
Table 4.10	Comparison of synthesis results for filters T1, T2, T3 and T4 using 0.18 μ m TSMC technology.....	73
Table 5.1	TDMA/PDC filter specifications....	80
Table 5.2	Implementation results of multi-band filter and masking channel filter.....	85
Table 5.3	Details of the number of channels extracted using proposed MSFRM based architecture	99
Table 5.4	Comparison of filter lengths in proposed MSFRM and SSFRM based channel filter	104
Table 5.5	Implementation results of MSFRM based channel filter	106
Table 5.6	Quantitative comparison of multiplication rate of a channelizer.....	115
Table 6.1	Specifications of the channels for the example using filter level reconfigurability	124
Table 6.2	Specifications of the channels for the example using architecture level reconfigurability	126
Table 6.3	Implementation results for a single channel extraction (Channel-4).....	127
Table 7.1	Synthesis result of product encoder.....	143

Table 7.2 0.18µm synthesis result of 200 tap D-AMPS channel filter.....146

Table 7.3 0.18µm synthesis result in comparison with reconfigurable FIR filters
[55-58]147

Table 8.1 Summary of the methods and results accomplished in this thesis153

List of Abbreviations

AFE	Analog Front End
ADC	Analog-to-Digital Converter
ASIC	Application Specific Integrated Circuits
AGA	Adaptive Genetic Algorithm
BCS	Binary Common Subexpression
BHCS	Binary Horizontal Common Subexpression
BHSE	Binary Horizontal Subexpression Elimination
BHSSE	Binary Horizontal Super Subexpression Elimination
BH	Bull-Horrock's
CSD	Canonical Signed Digit
CSE	Common Subexpression Elimination
CS	Common Subexpression
CRA	Contention Resolution Algorithm
CDMA	Code Division Multiple Access
CMFB	Cosine Modulated Filter Bank
DFTFB	Discrete Fourier Transform Filter Bank
DPU	Digit Processing Unit
D-AMPS	Digital Advanced Mobile Phone Systems
DAC	Digital-to-Analog Converters
DDC	Digital Down Conversion
DFE	Digital Front End
DSP	Digital Signal Processor
DF	Direct Form
DAG	Directed Acyclic Graph
DIM	Decimation-Interpolation and Masking
FPGA	Field Programmable Gate Array
FIR	Finite Impulse Response
FRM	Frequency Response Masking
FDF	Frequency Domain Filtering

FA	Full Adder
GA	Genetic Algorithm
GD	Graph Dependence
GSPS	Giga Samples Per Second
GFB	Goertzel filter bank
HCSE	Horizontal Common Subexpression Elimination
HCS	Horizontal Common Subexpression
HLS	High Level Synthesis
IIR	Infinite Impulse Response
IF	Intermediate Frequency
LD	Logic Depth
LO	Logical Operator
LUT	Look Up Table
LNA	Low Noise Amplifier
MSD	Minimum Signed Digit
MILP	Mixed Integer Linear Programming
MSFRM	Multi-stage Frequency Response Making
BHM	Modified Bull Horrock's
MSPS	Mega Samples Per Second
MCM	Multiple Constant Multiplications
MMCM	Multiplexed Multiple Constant Multiplication
MB	Multiplier Block
MAC	Multiply-Accumulate
MPRFB	Modified Perfect Reconstruction Filter Bank
MSFB	Multi-Stage Filter Bank
NR-SCSE	Non-Recursive Signed Common Subexpression Elimination
PSR	Peak Stopband Ripple
PC	Per-Channel
PDC	Personal Digital Cellular
PFT	Pipelined Frequency Transform
RF	Radio Frequency

ReMB	Reconfigurable Multiplier Block
RAG-n	Reduced Adder Graph-n-Dimensional
RNS	Residue Number System
SRC	Sample Rate Conversion
SA	Simulated Annealing
SDR	Software Defined Radio
SS	Super Subexpression
SSE	Super-Subexpression Elimination
TDMA	Time Division Multiple Access
TDF	Transposed Direct Form
TPFT	Tunable Pipelined Frequency Transform
TQMFB	Tree-structured Quadrature Mirror Filter Bank
VCSE	Vertical Common Subexpression Elimination
VCS	Vertical Common Subexpression
W-CDMA	Wideband Code Division Multiple Access

Abstract

The fundamental idea of software defined radio (SDR) is to replace most of the analog signal processing in the transceivers with digital signal processing in order to provide the advantage of flexibility through reconfiguration. The digital front-end of an SDR employs a filter bank channelizer to extract individual radio channels from the wideband input signal. Since the filter bank operates at the highest sampling rate in the digital front-end of the receiver, low power and high-speed architectures are required for its implementation. The compatibility of a filter bank channelizer with different communication standards is guaranteed by its reconfigurability. Realizing reconfigurable channelizers with low complexity, low power and high-speed is a challenging task. This thesis addresses the research problem of incorporating low power, high-speed and reconfigurability into the filter bank channelizer architecture for the wideband receiver of an SDR.

Filter banks based on finite impulse response (FIR) filters are commonly employed in SDRs. Generally higher-order FIR filters are required in channelizers to meet the stringent adjacent channel attenuation specifications of wireless communication standards. Coefficient multiplications in such higher-order filters consume huge power and chip area. Apart from low complexity, reconfigurability of these filters and filter banks is important to support multi-standard operation of SDRs. In this thesis, five new methods are proposed for realizing reconfigurable low power channel filters and filter banks.

The first method is a 4-bit binary representation-based CSE method called BCSE-4 for implementing higher-order low power FIR filters. The proposed BCSE-4 method reduces the filter complexity by eliminating the redundant horizontal subexpressions and super subexpressions. The critical path lengths (termed logic depth) of the coefficient multipliers realized using proposed BCSE-4 method are almost identical to those realized using conventional techniques. The proposed BCSE-4 method offers an overall adder reduction of 6.5% compared to the best known CSE algorithm called the binary

subexpression elimination algorithm, without any increase in critical path length of filter coefficient multipliers.

The second method incorporates reconfigurability with low complexity for the implementation of channel filter using multi-band filtering and frequency masking methods. The proposed channel filter is able to extract channels of multiple communication standards, using the same fixed-coefficient multi-band filter and reconfigurable masking filters to extract the desired channels according to the current mode of operation. The method offers complexity reduction of 51.2% and average power reduction of 12.7% when compared to per-channel (PC) approach.

As the third work, a reconfigurable filter bank and channel filter has been proposed, based on a multi-stage frequency response masking (MSFRM) technique. The basic idea is to interpolate a filter whose response has very wide transition-band and then to mask out un-desired frequency bands, obtained while interpolating the filter, by employing wide transition-band masking filters. MSFRM approach helps the user to enjoy large interpolation values, which caters to wide choices of bandwidth and centre frequencies of the multiple standard radio channels. The proposed reconfigurable filter bank allows the user to enjoy non-uniform bandwidth extraction, very narrowband channel extraction and multi-standard operation with low complexity, when compared to conventional filter banks in SDR channelizers. The proposed method offers an average gate count reduction of 38.5% and power reduction of 18.1% when compared to single-stage FRM.

In SDR channelizer, the channel filters need to extract radio channels that can have bandwidths which are related by fractional factors. Based on the FRM approach, a new channel filter has been proposed called as the decimation-interpolation and masking (DIM) method, which can extract channels, whose bandwidths are related by fractional factors, which forms the fourth contribution. The proposed channel filter offers reconfigurability at the architectural level and at the channel filter level and allows the user to have more control over the bandwidth and centre frequency when compared to

MSFRM approach, for extraction multiple standard radio channels. The proposed approach offers an average gate count reduction of 10.7% and has a power reduction of 5.9% over the PC approach.

A new reconfigurable, high-speed, low complexity FIR filter architecture for the masking filters of the MSFRM and DIM methods have been proposed, based on the residue number system (RNS). In contrast to conventional reconfigurable multiplier units used in previously proposed reconfigurable filter architectures, easily reconfigurable look up table based moduli multipliers has been employed in the proposed architecture. The proposed architecture has been synthesized on TSMC 0.18 μ m CMOS standard cell technology and gives 18.7% speed improvement when compared to conventional reconfigurable filters. To the best of knowledge, this is the first attempt towards integrating frequency response reconfigurability using RNS technique for implementing higher-order FIR filters.

All the proposed architectures in this thesis have been implemented on Xilinx Virtex 2v3000ffl152-4 FPGA using Lyrtech's ADAC signal master and tested using real-time inputs. Implementation results show that the proposed architectures offer good trade-off between low complexity and reconfigurability.

Chapter 1

Introduction

The demand for new wireless services requiring higher capacities and data rates have motivated the development of new generation cellular wireless communication systems. Emerging wireless communication systems are aimed at accommodating this demand through better resource management and improved transmission technologies. The coexistence of different cellular systems requires multi-mode, multi-band and multi-standard mobile terminals. The complexity and portability of wireless systems places a stringent requirement on the total power consumption of signal processing and communication systems. Providing low power while maintaining low cost is the primary challenge for communication systems as the carrier frequencies are approaching 2–3 GHz. Transceivers for multi-mode and multi-standard wireless communications are often implemented by replicating the receiver circuits for each standard. On the contrary, transceivers that can share circuit functions between different standards in an adaptive manner have the advantages of low power consumption, small chip area, and, most importantly, the potential for low cost. Although high level of integration is possible on silicon, the additional radio frequency hardware required to implement this type of multi-standard radio increases the total current consumption. The need for a multi-standard transceiver which can operate simultaneously for each independent standard, by reconfiguring the hardware, resulted in the evolution of Software Defined Radio (SDR) [1-6]. The interest in improving spectrum efficiency combined with the introduction of SDRs has created new intriguing possibilities for wireless radio researchers.

The term SDR signifies that, the hardware architecture can be reprogrammed to adapt with any wireless radio standard. SDR finds major application in mobile communication transceivers, generic cellular basestations, cognitive radio systems and military radio systems. Some of the benefits that will result with the realization of SDR are:

- Easier international roaming, enhanced flexibility for various services, improved personalization and choice for subscribers of mobile services.
- The potential to rapidly develop and introduce new value-added services and revenue streams with increased flexibility of spectrum management and usage for mobile network operators.
- The promise of increased production flexibility contributing to rapid production advancement for handset and basestation manufacturers.
- The prospect of increased spectrum efficiency resulting in better use of scarce spectrum resource for regulators.

The cellular basestations employ distinct receiver chains comprising of analog components such as analog mixers, local oscillators and analog-to-digital converters (ADCs) for each communication standard. Distinct chains of these analog components along with baseband processing units, linearly increases the complexity of the communication receiver with the number of received radio channels of different communication standards. But SDR based basestations employ a single analog and digital front end to receive the radio channels of all the communication standards. The digital front end, which does the entire signal processing tasks, is reconfigured to work for different cellular standards. Hence the cost of the analog part is independent of the number of received standards. The rapid transition of communications technology from analog to digital domain has been fulfilled with the introduction of SDR. This has enabled different communication functionalities such as digital up/down conversion, modulation/de-modulation, to be performed by software on an appropriate digital hardware. The use of such programmable digital hardware has improved the flexibility in communication receivers to a significant extent.

For SDR to become a reality, more research is needed in the areas of embedded systems, computing architectures and design methodologies to create an all-digital reprogrammable transceiver. The major research challenges in SDR can be grouped into the following areas:

- Advanced software technologies: Flexible models for computation, which allows combining event-driven and data driven applications [7]. Efficient implementations using a combination of hardware architectures which include

digital signal processors, general purpose micro-processors, field programmable gate arrays (FPGA) and new reconfigurable logic platforms.

- ADCs and Digital-to-Analog converters (DACs): - ADCs and DACs for an SDR are to be placed as close as to the antenna where they have to operate at the highest sampling rate. Research in this area aims realization of high-speed ADCs and DACs for sampling the wide dynamic range wireless signals. Currently, the highest speed of operation available for a commercial ADC is at the rate of 200 mega samples per second (MSPS) for a wordlength of 16 bits [8], which is still not the best for SDRs [9]. For a SDR, ADC sampling rate has to be at the rate of giga samples per second (GSPS) in order to directly digitize RF signals.
- Analog front end: - Analog front end consists of the antenna and the analog filter to cope with the radio frequency signals. The research in area of antenna extends to new advancements in wideband (fractal) antennas and electronically steerable (smart) antennas [11]. Analog filter research deals with tunable RF filters to suit the versatility of SDR concept [10,11].
- Digital front end (DFE):- The DFE of an SDR terminal replaces analog signal processing by digital signal processing. As the DFE comes immediately after the ADC, it has to operate at the highest sampling rate [5]. Due to the large bandwidth and dynamic range of the wideband input signal, the channel extraction needs to be implemented with low complexity and they have to be reconfigured for different wireless standards. Hence the complexity reduction, high-speed operation along with reconfiguration of DFE is a tricky task.
- Advanced baseband processing: This includes challenges in areas such as adaptive (smart) antennas, multi user detection, spread-spectrum processing, data coding and security [11].

The focus of the work presented in this thesis is on low power implementation of reconfigurable DFE of an SDR receiver. In contrast to a conventional radio transceiver, SDR uses the same hardware architecture that could be reprogrammed for any number of multi-standard channels by reconfiguring DFE. In addition to these reconfigurability requirements, the hardware employed for SDR must also meet the stringent power and speed specifications of the wireless communications systems.

1.1 Motivation

The DFE is a part of the SDR terminal where analog signal processing is replaced by digital signal processing [5]. The main functionalities of the DFE consist of channelization, digital down conversion and sample rate conversion. The channelizer in DFE is the most computationally demanding block of a SDR receiver, as it comes directly after ADC and hence needs to operate at the highest sampling rate. The channelizer extracts individual radio channels of wideband input signal from the output of ADC using a bank of digital filters, commonly known as *channel filters*. Digital filtering is the key function of the channel filter and it is achieved with the help of finite impulse response (FIR) filters due to its absolute stability and linear phase property. The main research issues in designing efficient channel filters are

- The channel filter needs to be implemented with low complexity due to the large bandwidth and dynamic range of the wideband input signal.
- The channel filter must also meet the stringent power and speed specifications to satisfy the resource-constrained battery operated mobile wireless communication systems.
- Sharp adjacent channel attenuation produces stringent requirements for the transition- bands of the channel filters, which would require large number of filter taps (coefficients).
- Multi-standard channel extraction in communication receivers requires channel filters to be reprogrammable. The channel filters must be reconfigurable, in order for the bandpass filters to dynamically adapt to the new centre frequencies and bandwidth with minimum computational overhead (low power) and high-speed.

Various architectures are proposed in literature for efficient implementation of FIR filters. For SDR, reconfigurable filters are required and they must be further optimized or new architectures need to be developed to meet the stringent area, power and speed specifications of the channelizer in an SDR. Another key requirement in SDR is the reconfigurability of the channelizer to work with multiple wireless communication standards. The conventional method of achieving reconfigurability is by switching the operation among distinct receivers based on the current mode of

operation. This is not an efficient approach as complexity and power consumption will be high.

Realizing reconfigurable channelizers with low complexity is a challenging task. The motivation behind the work in this thesis is to deal with these research issues related to the realization of low power reconfigurable digital filter banks, which have not been sufficiently addressed in literature.

1.2 Objectives and Contributions

This thesis addresses the design and realization issues of low power reconfigurable and hardware-efficient digital filter banks for SDR receivers. The main objective of the thesis work is to realize low power, low complexity, reconfigurable digital filter banks for multi-standard communication receivers. The main tasks are as follows

- To propose efficient realizations for low power and high-speed channel filters by eliminating redundancy in coefficient multipliers. Coefficient multiplication is the most expensive operation in digital filters and filter banks as it consumes most of the area, power and time of filtering operations. The number of additions or (subtractions) used to implement coefficient multiplication determines the hardware complexity of the filter. Hence the primary objective is to minimize the number of adders needed to realize the filter coefficient multipliers.
- To incorporate reconfigurability to the channel filters for multi-standard channel extraction in wireless communication receivers.
- To propose multi-standard reconfigurable filter bank channelizer. In particular, this thesis aims to design and realize multi-standard filter bank architectures in order to achieve low power and low complexity wireless communication receivers.
- To implement the proposed architectures using FPGA methodology and comparison of synthesis results with existing methods.

The contributions of this thesis are:

A new 4-bit binary-based common subexpression elimination (BCSE-4) method based on binary representation of filter coefficients which employs 4-bit common subexpressions for implementing higher order low power FIR filters with fewer number of adders than canonical signed digit based common subexpression elimination (CSE) methods has been proposed. The critical path lengths (termed logic depth) of the coefficient multipliers realized using proposed BCSE-4 method are almost identical to those realized using conventional techniques. Thus the proposed CSE method produces filters with low complexity coefficient multipliers without increasing the delay.

Incorporating reconfigurability with low complexity for the implementation of channelizer is a challenging task. To tackle this, a reconfigurable channel filter architecture using multi-band filtering and frequency masking methods is proposed. The proposed channel filter is able to extract channels of multiple communication standards, with different bandwidths and different stopband attenuations using a fixed-coefficient multi-band filter and reconfigurable masking filters according to the current mode of operation.

A reconfigurable filter bank and reconfigurable channel filter have been proposed based on a multi-stage frequency response masking (MSFRM) technique. The basic idea is to interpolate a filter whose response has very wide transition-band and then to mask out un-desired frequency bands, obtained while interpolating the filter, by employing wide transition-band masking filters. MSFRM approach enables the user to have extensive range of interpolation values, which caters to wide choices of bandwidth and centre frequencies of the multiple standard radio channels. All the filters in MSFRM technique are having wide transition-band for any arbitrary interpolation value and hence they are lower order in nature and less complex. The proposed reconfigurable filter bank allows the user to have non-uniform bandwidth extraction, very narrowband channel extraction and multi-standard operation when compared to conventional filter banks in SDR channelizers. The proposed channel filter and filter bank have the flexibility of dynamically changing of frequency responses by changing the interpolating values, with very low complexity.

In SDR channelizer, the channel filters need to extract radio channels that can have bandwidths which are related by fractional factors. Channels with bandwidth related by fractional factors can be extracted using the proposed decimation – interpolation and masking (DIM) technique. The fractional bandwidth extraction is made possible by using a two-stage approach, where the first stage of the proposed architecture employs a low pass linear phase FIR filter, whose frequency band edges can be varied by changing the decimation and interpolation factors. The proposed channel filter offers reconfigurability at the architectural level and at the channel filter level and allows the user to have wider control with respect to MSFRM approach, over the bandwidth and centre frequency, for extraction multiple standard radio channels.

Out of the three methods multi-band filtering and masking method will have the highest complexity, as the method makes use of a multi-band filter realized for the most stringent specification. Multi-band filtering and masking method is useful when the channels to be extracted are scarcely located in a wideband frequency spectrum. The advantage of multi-band filtering and masking method is that, absolute control over the passband width and location of centre frequency is possible, with prior knowledge of the wideband spectrum. MSFRM based channel filtering gives the least complexity solution for channels to be extracted whose bandwidth is $1/L$ (L = interpolation value) of the modal filter bandwidth. DIM based channel filtering extracts channels whose bandwidth shares a fractional relation (D/L , D = decimation value and L = interpolation value) with the modal filter bandwidth. This extra reconfigurability property comes on the cost of higher complexity for the modal filter when compared to MSFRM based approach.

The above techniques 2, 3 and 4 require reconfigurable masking filters to mask out the channel-of-interest. A new reconfigurable, low complexity FIR filter architecture for SDRs have been proposed, based on the residue number system (RNS). The method focuses on direct implementation of RNS-FIR filters using look up tables (LUTs) and thus results in high-speed operation. In contrast to conventional reconfigurable multiplier units in filter architectures, easily reconfigurable LUT based moduli multipliers has been employed in the proposed architecture. For a given moduli set, the architecture can be reconfigured for multiple standards with varying number of taps. The proposed architecture has been synthesized on TSMC 0.18 μ m CMOS

standard cell technology and gives superior results when compared to conventional reconfigurable filters. To the best knowledge, this is the first attempt towards integrating reconfigurability using RNS technique for implementing higher-order FIR filters.

All the proposed architectures have been implemented on Xilinx Virtex 2v3000ff1152-4 FPGA using Lyrtech's ADAC signal master and tested using real-time inputs.

1.3 Outline

This thesis is organized as follows: Chapter 2 presents the literature review of various functionalities and research challenges in the SDR channelizer along with a detailed overview on filter banks. In Chapter 3, the background knowledge pertaining to the design and optimization of different channel filter architectures and its reconfigurability issues are presented. Chapter 4 presents the proposed 4-bit binary common subexpression elimination (BCSE-4) algorithm. The proposed multi-band filtering and masking is presented in Chapter 5. Chapter 5 also discusses the proposed multi-stage frequency masking channel filter and filter bank architecture. In Chapter 6, channel filter realization based on decimation, interpolation and frequency response masking technique is presented. Chapter 7 presents the proposed reconfigurable RNS-FIR filter architecture. This thesis is concluded with an account of its contributions and possible future directions in Chapter 8.

Chapter 2

Software Defined Radios: A Literature Survey

In this chapter, the basic concept of the software defined radio (SDR) receiver is presented. The emphasis of this chapter is to provide a good understanding about the functionalities of various parts of the receiver architecture. A review of the conventional channelizer architectures along with their complexity aspects is also provided.

2.1 Definition and Objectives of SDR

The definitions of SDR found in literature [1-4] can be summarized as follows.

- Flexible transmitter (TX)/ Receiver (RX) controlled and programmable by software.
- Signal processing able to replace, as much as possible, radio functionalities.
- “Air-interface Downloadability”: Radio equipment dynamically reconfigurable by downloadable software at every level of the protocol stack.
- Software realization of terminals “Multiple mode/standard”.
- Transceiver, where the following can be defined by software.
 - Frequency band and radio channel bandwidth.
 - Modulation and coding scheme
 - Radio resource and mobility management protocols.
 - User applications.

These parameters can be adapted and changed by the network operator, the service provider, or the final user. Integrating all the above, the SDR can be defined as, *“Software radio is an emerging technology, thought to build flexible radio systems, multi-service, multi-standard, multi-band, reconfigurable and reprogrammable by software”* [2]. The flexibility of an SDR system is based on its capability to operate in multi-service environments, without being constrained to a particular standard, but able to offer, in theory, services of any already standardized systems or future ones on any radio frequency band. For realizing an SDR that satisfies this definition, many

challenging research issues need to be addressed. The two main goals that need to be achieved are [1- 4]:

1. To move the border between analog and digital domains as much as possible toward radio frequency (RF), and achieving this aim by accomplishing the ADC and DAC conversions as near as possible to the antenna.
2. To replace application-specific integrated circuits (ASICs) with programmable digital signal processors (DSPs) or FPGAs for intermediate frequency (IF) processing, in order to define as many radio functionalities as possible in software.

The primary target of the first goal was to extend the digital domain at the IF stage, keeping the RF stage analog. The final target is to have a totally digital transceiver. The replacement of ASIC technology with DSP/FPGA opens the way in two possible horizons [2]:

1. Software implementation of IF and baseband functions, such as coding, modulation, channelization, equalization, and pulse shaping.
2. Reprogrammability of the system to guarantee multi-standard operation.

2.2 SDR Receiver Architecture

Figure 2.1 Ideal software defined radio receiver.

The software radio receiver illustrated in Fig. 2.1 has been defined as ideal because there are several aspects that make it, at the moment, far from realizable. An ideal SDR receiver should be capable of receiving ultra-wide bandwidth of RF signals (hundreds of MHz to units or tens of GHz), interpreting many given air-interface radio standards using software. In order to do so, the ADC should be “shifted” as close as possible to the receiver’s antenna and should have very good control to reduce the

jitter effects in analog-to-digital (A/D) conversion, as the conversion is directly at RF. Another criterion is that more number of RF stages may be needed in order to cater an ultra-wide multi band input, depending on the radio band used for the SDR system. The ideal SDR receiver contains minimal analog signal processing.

An ideal SDR receiver typically consists of anti-aliasing filter, low noise amplifier, ADC and digital signal processing unit. In Fig. 2.1, the anti-aliasing filter band-limits the input signal to prevent aliasing while sampling the signal. This is followed by a low noise amplifier (LNA) that amplifies the input signal to a desired level. These two comprises the RF stage. The signal is converted to digital form using ADC. Once the signal is converted to digital form, the digital signal processing unit will perform all the necessary signal processing tasks which were earlier achieved using analog circuits. ADC and signal processing unit constitute the Baseband (BB) stage. In an ideal SDR, since all the channel separation tasks are performed digitally, the analog front end and the ADC must process the total signal bandwidth for which the receiver is designed.

However, there are some key issues still prevent the realization of an ideal SDR receiver.

- In order to transmit and receive wideband frequencies wideband antennas are required. But wideband antennas will be less performing and less cost effective than narrowband antenna as the antenna size is proportional to operating wavelength [9]. Hence the antenna that ideally receives and transmits wideband of frequencies is not realizable with currently available technology.
- Another problem stems from the ADC bottleneck. As first envisaged by Mitola [1], all RF and baseband signal processing is digital, enabled by an ADC at the antenna (Fig. 2.1). The ADC in the SDR receiver for such a case must fulfill extraordinary specifications. For example, to digitize the frequency band from 800 MHz to 5.5 GHz, where all of today's cellular and wireless local area network (WLAN) channels lie, will require a 12 bit, 11 Giga-samples per second (GSPS) ADC [11]. State of the art ADCs reach about 3 GSPS, and also this with a relatively low 8-bit resolution (e.g., National Semiconductor's ADC08D1500). Currently, for a precision of 16 bits, the highest speed of operation available for a

commercial ADC (Texas Instruments ADS5485) with 200 Mega samples per second (MSPS) [8]. According to the Nyquist-Shannon sampling theorem, an input signal should be sampled at a rate, which is at least twice the RF bandwidth of the signal, in order to be able to reconstruct it. Also the dynamic range of RF signal is very high. Therefore digitization of RF signal is beyond the scope of the modern day ADCs as its dynamic range and conversion are known to progress at a rate much slower than Moore's law [11, 12].

- Other fundamental limitations of ADC due to its non-ideal nature are low resolution (quantization error), non-linear behavior, deviation from accurate sample timing intervals (jitter error) and noise, which limit its performance [12, 13].
- The final stumbling block to realize an ideal SDR is the limitation of computation power. Assuming that a perfect ADC exists, the amount of digital information (samples) to be processed by the digital signal processing unit(s) surpasses the computation capacity of presently available computing platforms and might require Giga-FLOPS performance [1].

Thus the ideal SDR receiver shown in Fig. 2.1 is not realizable in practice with today's technology. A more feasible SDR receiver architecture is shown in Fig. 2.2.

Figure 2.2 A feasible software radio receiver.

From the above discussion, it can be concluded that only partial band digitization is possible in a feasible SDR receiver in contrast to full band digitization envisaged for an ideal SDR. Limiting the bandwidth (BW) of the receiver makes it possible to devise a suitable antenna, to alleviate ADC sampling rate limitation, to reduce jitter error and quantization error and to decrease computation load. Therefore, IF stage is introduced in order to deal with ADC bandwidth limitations. Hence for the feasible SDR, in order to cover all the services to be supported by the SDR, a limited band has to be selected out of the full band by means of analog conversion and IF filtering. In Fig. 2.2, the LNA and anti-aliasing filter are same as in Fig. 2.1. The mixer is used to down convert the frequency of the input signal from RF to IF so that today's ADCs can easily digitize the IF signal. The portion of the SDR which includes LNA, mixer and anti-aliasing filter is known as analog front end. The main function of the analog front end is to reduce the frequency from RF to IF and to band-limit the input signal to prevent aliasing.

The portion of the SDR receiver that comes immediately after the ADC is referred to as the digital front end (DFE) [5]. The DFE is the most computationally demanding block of a SDR receiver as it operates at the highest sampling rate. The DFE is likely to be implemented in FPGAs and ASIC, which provide a trade-off between performance and flexibility. The DFE performs digital down conversion (DDC), sample rate conversion (SRC) and channelization as shown in Fig. 2.2, which will be explained in more details in Section 2.3.2. The output of DFE will have individual radio channels at baseband extracted from the IF input signal. The DFE, which comes directly after the ADC, needs to operate at IF. Thus the DFE cannot be realized efficiently using DSPs because of speed constraints associated with DSPs. Hence dedicated hardware architectures are required for implementing DFE [5]. Once all the channels are obtained at baseband, at the output of DFE, they can be further processed using DSPs.

2.3 SDR Receiver Functionalities

The critical functionalities of the SDR receiver architecture are explained in this section. The details of the functionalities of the ADC and the DFE are presented.

2.3.1 Analog-to-Digital conversion

The analog-to-digital conversion performs sampling and digitization of the wideband input signal. The two basic approaches to the analog-to-digital conversion are described below [9, 14].

1. *Full Band Digitization*: In this case, the whole bandwidth containing channels of all services need to be digitized. This bandwidth can be over 100 MHz which means the dynamic range of ADCs have to be larger than 100 dB. Even though full band digitization is the best solution, it is not efficiently realizable, with respect to power and cost, with today's ADC technology.
2. *Partial Band Digitization*: In this case, only the bandwidth equal to the widest channel bandwidth of all the supported services needs to be digitized. This can be easily achieved using currently available ADCs.

Recently, different implementation technologies have been experimented to improve the performance of ADCs [9]. Though the semiconductor approach has been the forerunner for ADC implementation technology, the recent trend is to employ alternative technologies like optical sampling and super conductor technology. In semiconductor technology, the resolution degrades by 1-bit for every doubling of the sampling rate. An analysis of signal to noise ratio shows that 1-bit/octave slope is related to sample variation called aperture jitter. The aperture jitter can be reduced to a significant extent by employing optical sampling. The main advantage of an optically sampled ADC lies in the low timing jitter of the mode-locked laser source. The other breakthrough technology that has opened the doors for high-speed ADCs with improved resolutions is the rapid single flux quantum (RSFQ) [9, 14]. This technology is based on a fundamental quantum mechanical property of super conductors, stating the existence of magnetic flux in discrete quantized form. In this technology single flux quantum pulses represents binary value. The speed and resolution comparison of different ADC technologies are shown in Table 2.1 [9, 83]. From Table 2.1, it can be concluded that, it is basically a trade-off between technology, speed of operation and resolution to select the best ADC solution for SDR receivers.

Table 2.1
Comparison of ADC technologies.

ADC Technology	Resolution	Speed	Status
Semiconductor Based	6 bits	3.2 GS/s	Commercially Available
	10 bits	1 GS/s	Commercially Available
	4 bits	35 GS/s	Experimental
Optical Sampling	8.2 bits 12 bits	505 MS/s Multi GS/s	Experimental Proof of Concept
Superconductor RSFQ	11 bits	175 MS/S	Experimental

2.3.2 Digital Front End

The DFE is a hardware platform for realizing front-end functionality of SDR by means of signal processing. The DFE performs digital down conversion (DDC), channelization and sample rate conversion (SRC) as shown in Fig. 2.2 and Fig. 2.3. Fig. 2.3 shows the detailed functionalities of the DFE.

Figure 2.3 Digital Front End.

The DFE must deliver a digital signal which is ready for baseband processing, with a sample rate determined by the currently operating air-interface. This digital

signal represents the channel-of-interest of bandwidth, B_i centered at $f_c = 0$. In Fig. 2.3, the wideband input signal of three non-uniform channels is shown and each of them represents three signals of different bandwidth. The DDC down-converts the wideband signal so that the channel-of-interest is at baseband. The channel filter isolates the channel-of-interest from the adjacent channels. The SRC performs sample rate change such that the output of DFE should be an integer multiple of the baud rate. A brief analysis of the three stages of operation (DDC, Channelization and SRC) in the DFE is presented in the following sections.

2.3.2.1 *Digital Down Conversion (DDC)*

SDR receivers often have fast ADC delivering vast amounts of data; but in many cases, the channel-of-interest represents a small proportion of that bandwidth. A DDC allows the rest of that data to be discarded, allowing more intensive processing to be performed on the channel-of-interest. When partial band digitization is employed, the basic channel selection task is performed in the analog domain by converting the signal of interest to a fixed IF. Hence DDC can be done using relatively simple hardware. The two parameters that influence the effort of implementing DDC are the frequency value of IF, f_{IF} , and sample rate f_s . A special case is the center frequency of the digitized channel-of-interest equals a quarter of the sample rate. This is achieved if the center frequency of the channel-of-interest before digitization equals to an odd multiple of a quarter of sample rate as given in (2.1)

$$f_{IF} = \frac{nf_s}{4}, \quad n=1, 3, 5, \dots \quad (2.1)$$

Then DDC can be performed by multiplying the signal with the sequences $[0 \ 1 \ 0 \ -1]$ and $[1 \ 0 \ -1 \ 0]$, representing the digital sine and cosine-signal, respectively, at a quarter of the sample rate. Hence, DDC operation can be done using a lookup table containing samples of a sinusoidal carrier. The lookup table replaces the local oscillator used in an analog down convertor. Given the constraint of equation (2.1), digitization and down conversion of bandpass signals can be combined [15]. Basically the task of bandpass analog-to-digital conversion is followed by an I-Q down conversion as shown in Fig. 2.4.

Figure 2.4 Bandpass ADC followed by digital down conversion.

2.3.2.2 Channelization

Channelization comprises all tasks necessary to select the channel-of-interest. This includes channel filtering and de-spreading. Channel filtering is necessary to extract frequency-divided channels. Thus, the filters have to attenuate adjacent-channel interferers and have to meet the blocking characteristics of the current mode of operation. De-spreading is required for final user channel selection in the case of spread-spectrum systems such as wideband code-division multiple access (W-CDMA). De-Spreading is the functionality where in spread-spectrum systems, the task of de-correlation and sample rate conversion to symbol rate is realized. For the realization of channelization, the speed of today's DSPs is not sufficient enough to cope up with the high sampling rate requirements. Hence dedicated hardware has to be designed. A detailed overview of channelizer is given in Section 2.4 which is the focus of this thesis.

2.3.2.3 Sample Rate Conversion (SRC)

An SDR must cater different mobile communication standards and each of the standards has to process signals at different sample or symbol rates. In order to provide signals at symbol or chip-rate (i.e. an arbitrary target rate), SRC has been introduced as one of the functionalities of the DFE of a SDR transceiver. Sample rate adaptation involves undersampling of the signals at the output of the channel filters to match the sample rate to the selected channel bandwidth, which is the narrow band signal, compared with wide spectrum input signal. SRC is the task of converting the sample rate of a digital signal to another sample rate while a certain amount of information,

usually in a limited frequency band, must be preserved [16]. The sample rate can be either made adaptive to different standards or fixed followed by a digital sample rate conversion. Basically two approaches are possible for sample rate conversion [16]:

1. Both f_{IF} and f_s are made parameterizable by keeping the ratio between them according to the equation (2.1). In order to simplify the down conversion process, each signal can be digitized with the clock rate of the standard of current operation.
2. Digitization is done with a clock rate fixed at f_{IF} , sample rate conversion can be performed by means of mathematical interpolation. This can be implemented in different ways:
 - a) The straight forward way of performing sample rate conversion by a rational factor, is to interpolate and decimate the signal by integer factors. But if no proper image and alias rejection is done, this method results in improper results especially for signals with high dynamic ranges. Hence this approach results in expensive implementations.
 - b) By reducing the dynamic range of the signal by means of sharp cut-off channel selection filters, the cost of approach in (a) can be reduced.
 - c) Assuming a block-wise processing, a certain number of samples per block can be dropped in order to reach the desired sample rate. The error thus introduced should be cancelled after the dropping or the signal has to be pre-distorted in a way that the dropping leads to a distortion-free signal. This approach is referred to as ‘asynchronous decimation’.

In the case where error cancellation can be performed efficiently, the dropping process is most elegant and straight forward [16].

2.4 Channelization for SDR Receivers

Channelization is a process where single, few, or all channels from the frequency band of a wideband input signal are separated for further processing by bandpass digital filters known as *channel filters*. The separation of single channel is usually done by down-conversion followed by channel filtering and optional sample-rate conversion. The channels of interest may be of equal or different bandwidths and may be uniformly or non-uniformly, continuously or non-continuously distributed over the

input frequency band. Channelization is the first stage of digital signal processing where parameters such as bandwidth and sample-rate are changed according to the current mode of operation. For SDR, it is desirable to realize as much of the channelization functionality as possible with DSP. In the case of a mobile handset, it needs to operate at different channels of interest according to the various communication standards depending upon the geographical location. The power consumption of the channelizer is an important aspect for the battery powered mobile handset whereas the base station has relaxed power constraints. The ultimate aim of SDR is to realize transceiver functionalities on battery powered mobile handsets, where the system needs to be power efficient. The basic assumptions of channelization are [17, 18]:

- The analog front end employs complex I-Q downconversion.
- All channels can have the same bandwidth (i.e., stem from the same air-interface) or can have different bandwidths (i.e., simultaneous reception of channels of different air-interfaces).
- The bandwidth of the channels must be variable/tunable (i.e., software definition of the channel bandwidth).
- The bandwidth, B , of the digitized signal is as large as possible. This implies that the number of channels of interest, N_I , is usually very small compared to the number of channels which the bandwidth B comprises.

Two key issues that determine the technical requirements of the channelizer are the spectral content of the wideband channel-of-interest and the types of processing devices available for channelization [19]. The technical requirements for the channelization approach are driven by the frequency allocation plans supported by the SDR. The system architecture of the cellular communications like GSM900 with an uplink band from 890 to 915 MHz and a downlink band from 935 to 960 MHz, defines a fixed carrier spacing with a constant RF bandwidth per carrier channel [20]. Both of these bands contain 124 carrier channels spaced 200 kHz apart. The channelizer supporting this type of network may be able to utilize the redundancy of the channel structure to provide an efficient channelization mechanism. On the other end, both the carrier frequency and RF bandwidth per carrier are dynamically assigned as found in a

multi-standard communications system such as a multi-standard satellite gateway [21]. The processing devices for channelization must be capable of operating at high speed. Hence normally, front-end channelization is typically done using FPGAs, due to high-speed requirements in dealing with the wideband input, and back-end (baseband) processing, which is performed on a per-channel basis, is done using digital signal processors or general purpose processors.

In SDR based mobile handsets, a reconfigurable channel filter is employed, where the filter is reconfigured to the current standard of operation. This is because only one-channel is of interest at a time in mobile handsets. However in SDR base-stations, several channels have to be extracted in parallel. An obvious approach to perform this task is to have a bank of channel filters for each channel. This approach is known as the per-channel approach. An alternative is to employ digital filter banks. The basics of these approaches are discussed in the following subsections.

2.5 Channelization Architectures

2.5.1 The Per-Channel Approach

Per-Channel (PC) approach of channelization is a straight forward implementation, to use a single-channel channelizer for each channel-of-interest, and connect them all to the input signal. The principle of PC approach is basically a parallel arrangement of many-to-one channel channelizer. Fig. 2.5 illustrates the architecture of M-channel PC approach. In conventional PC approach, each cellular standard would require a separate channelizer and each channelizer must perform digital filtering, DDC and SRC for further baseband processing [17]. In Fig. 2.5, the order of channelization is filtering ($H_0(z)$ to $H_{M-1}(z)$), DDC, SRC and finally baseband processing. In Fig. 2.5, $H_0(z)$, is a lowpass filter and all other filters ($H_1(z)$ to $H_{M-1}(z)$) are bandpass filters. Another approach is to perform DDC followed by filtering and in this case, all the filters become lowpass filters (all filters are $H_0(z)$). Both these cases are discussed below.

Fig. 2.5 shows the first case where bandpass filtering is followed by DDC. The digitized IF signal is filtered using the complex valued bandpass filter; the result of this filtering operation is finally down converted to baseband. The impulse response of

the bandpass filter is complex valued and the input signal is also complex valued. Thus each addend requires $4N$ multiplications per output sample (for an FIR filter with N coefficients).

In the second case, DDC is performed to the wideband signal at the output of ADC to bring the channel-of-interest to baseband. Lowpass filtering operation is then done to remove the adjacent channel interferers. Though both solutions are equivalent in terms of their input-output behaviour, there are differences in implementations [17]. In the case of lowpass filtering after DDC, the impulse response of the lowpass filter is real-valued. Hence, with a complex valued input signal, each addend only requires only $2N$ multiplications per output sample. The multiplication per output sample is only half of that of the approach where bandpass filtering is followed by DDC. Therefore for the PC approach, lowpass filtering after DDC is preferred.

Figure 2.5 Principle of PC approach for channelization.

The PC approach provides a great deal of flexibility in the choice of channels to be separated. Each single-channel channelizer can be individually designed for a particular communication standard. Furthermore, the separated channels are not constrained to be of the same bandwidth or to be uniformly distributed over the frequency input band. However, once such a channelizer is designed, it lacks flexibility to accommodate signals of new standards. The major drawback of the PC approach is that the complexity of the digital part of the wideband receiver increases linearly with the number of channels. For wideband receivers, where many channels

are to be separated, silicon costs and power consumption are extremely high for PC approach which led to the development of widely employed discrete Fourier transform filter banks (DFTFBs) [17].

2.5.2 The Filter Bank Approach

An alternative to PC approach is the filter bank approach. According to multi-rate signal processing theory, the filter bank channelizer can extract every channels between $[(-f_s/2), (f_s/2)]$, where f_s is the sampling frequency of the wideband ADC. The complexity of the filter bank channelizer is constant, i.e., independent of the number of received channels. Therefore, when large number of channels is needed in a receiver, the filter bank channelizer is computationally efficient.

Filter bank channelizer is generally called as *modulated filter bank*, as all the filters are derived by modulating (i.e., frequency shifting) a lowpass prototype filter. The modulation can be, discrete Fourier transform (DFT), or the cosine transform. If the center frequencies f_k , where $k=0, 1, \dots, (K-1)$ of the K filters are evenly distributed over the available frequency band $(0, f_s)$ where f_s is the sample rate, e.g.

$$f_k = \frac{k}{K} f_s \text{ where } k = 0, 1, 2, \dots, K-1 \quad (2.2)$$

the filter bank is called a *uniform filter bank*. In a uniform filter bank, the centre frequencies f_k of all K filters are equally distributed. The channel stacking arrangement shown by Equation (2.2) is called an even channel stacking ($f_0 = 0$). The modulated filter bank reduces the overall cost (hardware cost and multiplication rate) for implementing the prototype filter and the modulation operation when compared to the parallel arrangement of K independent filters.

The uniform DFT filter bank (DFTFB) is realized by implementing one lowpass filter and correspondingly modulating it with DFT. The DFTFB, a classic example of filter bank approach, is developed as an efficient substitute for PC approach when the number of channels to be extracted is large and these channels are of uniform bandwidth. A typical application scenario of DFTFB is extraction of multiple channels of a single communication standard in a basestation receiver. The main advantage of DFTFB is that, it can efficiently utilize the polyphase decomposition of filters. The basic block diagram of DFTFB, shown in Fig. 2.6, where $H_0(z), H_1(z), \dots, H_{K-1}(z)$ represents the polyphase representation of the lowpass

prototype filter. The downsampling factor K can be brought before the filtering operation, which effectively reduces the sampling rate at which prototype filter has to operate. The channel separation is equal to the input sample rate (f_s) divided by the DFT size (K) as shown in equation (2.3). Hence DFTFBs provide fixed channel stacking. The output sample rate is a fraction of the input sample rate.

$$\Delta f_{channel} = \frac{f_s}{K} \quad (2.3)$$

Figure 2.6 DFT filter bank.

Efficient implementations of a channelizer using DFTFBs are available in literature [17]. It can be seen from Fig. 2.6 that DFTFB can be realized by implementing one lowpass filter and a corresponding modulator such as inverse DFT (IDFT). Thus instead of implementing K separate channel filters as in the case of PC approach, a single lowpass filter followed by DFT is only required (complexity of IDFT is same as that of DFT), provided all the channels of the input signal have equal bandwidth. However, DFTFBs have following limitations for multi-standard SDR receiver applications [17, 19]:

1. DFTFBs cannot extract channels with different bandwidths. This is because DFTFBs are modulated filter banks with equal bandwidth of all bandpass filters. Therefore, for multi-standard receivers, where channel bandwidths are distinct for each standard, distinct DFTFBs are required for each standard. Hence the complexity of a DFTFB increases linearly with the number of received standards.
2. Due to fixed channel stacking, the channels must be properly located for selecting them with the DFTFB. The channel stacking of a particular standard

depends on the sample rate and the DFT size. To use the same DFTFB for another standard, the sample rate at the input of the DFTFB must be adapted accordingly. This requires additional SRCs, which would increase the complexity and cost of DFTFBs.

3. If the channel bandwidth is very small compared to wideband input signal (extremely narrowband channels), the prototype filter must be highly selective resulting in very high-order filter. As the order of the filter increases, the complexity increases linearly. Also the DFT size needs to be increased.

From multi-standard wireless communication perspective, the major drawback of the DFTFB approach [17] is that, it cannot perform channelization for channels with different bandwidths. In order to extract channels for various bandwidths, the prototype filter has to be reconfigured. Hence *reconfigurability* is a key requirement in SDR receivers. Ideally, the reconfigurability of the filter bank must be accomplished by reconfiguring the same prototype filter in the filter bank to process the signals of the new communication standard with the least possible overhead, instead of employing separate filter banks for each standard. However reconfiguration of DFTFB suffers from following overheads:

1. The prototype filter needs to be reconfigured. Generally DFTFB employs the polyphase decomposition. Hence reconfiguration can involve changing the number of polyphase branches and the number of coefficients in each polyphase branch which is a tedious and computationally expensive task.
2. Downsampling factor needs to be changed. If downsampling is to be done after filtering, then separate DDCs are required, which will increase the cost of implementation.
3. The DFT needs to be reformulated according to the number of polyphase branches, which is also expensive.

For example, if switching from a 8-channel filter bank to 16-channel filter bank is considered, the number of polyphase branches needs to be changed from 8 to 16 (first limitation of DFTFB), the downsampling factor needs to be adjusted from 8 to 16 (second limitation of DFTFB) and the 8-point DFT needs to be expanded to 16-point DFT (third limitation of DFTFB).

2.5.3 The Goertzel Filter Bank (GFB)

Figure 2.7 Goertzel filter bank.

In order to overcome the fixed channel stacking limitation of DFTFB, a filter bank for which the response of the prototype filter can be modulated to any center frequency is needed. This can be done by using the Goertzel algorithm [22] as a substitute to DFT operation. Hence, to overcome the problem of fixed channel stacking in DFTFBs, the Goertzel filter bank (GFB) architecture was proposed in [17]. In GFB, as shown in Fig. 2.7, the DFT operation is replaced by a modified Goertzel algorithm which performs the modulation of the prototype lowpass frequency response to any centre frequency. The advantage of Goertzel algorithm is that it calculates only a single point of a DFT and thus offers a low complexity solution. But the GFB is also a type of modulated filter bank; hence it cannot extract channels with different bandwidths, as in the case of DFTFB. Also, extraction of narrow-band channels using GFB requires a very narrow passband prototype filter, which is expensive. The GFB approach requires infinite impulse response (IIR) filter for the implementation of Goertzel algorithm. The stability of IIR filters are determined by the poles. While reconfiguring the filter bank from one communication standard to another, IIR filters can become unstable because of the new pole values after reconfiguration. Even though a theoretical introduction of GFB as a solution to some of the problems of DFTFB is given in [17], there is no consideration on the actual implementation complexities of GFB.

2.5.4 The Pipelined Frequency Transform (PFT)

The Pipelined Frequency Transform (PFT) algorithm [23] is based on a binary tree of DDCs and SRCs (Fig. 2.8) where units of DDC followed by SRC are used for dividing

their input band into two half-bands with half sampling rate. This algorithm creates a binary tree that splits the input frequency in two half-bands and then splits each half-band again into two half sub-bands and so on, until the last tree level produces the required separated channels. Using this algorithm single-channel channelizer complexity can be reduced dramatically, taking advantage of filters symmetry and restricting the output sample rate to be quarter of input sample rate in each single node in the tree. Observing that the components in each stage operate at half the sampling rate of its preceding stage components, a considerable hardware optimization can be performed. However the channels are outputted serially, and therefore some extra processing is required for distributing them in distinct outputs. The PFT architecture is more economical in terms of silicon use and power consumption when compared to PC architecture, especially when many channels are to be separated from the frequency input band [23].

Figure 2.8 The pipelined frequency transform architecture.

In the PFT approach, the problem with power-of-two subband stacking can be overcome by a tunable PFT (TPFT) architecture [24]. The TPFT algorithm is an adaptation of the PFT that alleviates this inflexibility by introducing interleavers that provide intermediate outputs from the PFT stages that may be used for fine tuning channelization [24]. These interleavers will help in fine tuning of channelization process and thus add more flexibility to the PFT architecture. Thus in TPFT, two levels of tuning are done, a coarse tuning at the PFT level and a fine tuning using another complex up/down converter assisted by a numerical controlled oscillator.

However the implementation complexity of TPFT approach is much more than that of the PFT approach. Hence TPFT is not a hardware-efficient solution for wideband channelization.

2.5.5 The Tree structured Quadrature Mirror Filter Bank (TQMFB)

A reconfigurable channelizer using tree-structured quadrature mirror filter bank (TQMFB) has been proposed in [25]. The TQMFB consists of a tree of quadrature mirror filter banks, splitting the frequency band of input signal into high and low frequencies at quadrature frequency in each stage. Thus the TQMFB approach in [25] is very similar to the PFT approach in [23]. The desired channel is obtained at an appropriate stage corresponding to the bandwidth of the channel-of-interest. The main drawback of the TQMFB is its delay in obtaining the desired output due to multi-stage filtering and decimation processes. The channelizers in [23] and [25] suffer from the drawback that they can only extract signals whose channel spacing are related by a factor of two. This constraint is imposed by the power-of-two subband stacking adopted in these architectures. Another problem of the methods in [23, 25] is that, as the subband decomposition tree extends, the wordlength of the output increases linearly and finite wordlength multiplication would introduce truncation error, which propagates along the tree.

2.5.6 The Modified Perfect Reconstruction Filter Bank (MPRFB)

Figure 2.9 Modified perfect reconstruction filter bank.

In [26], an efficient method called as modulated perfect reconstruction filter banks (MPRFB) is proposed for channelizing frequency division multiplexed (FDM) channels in received wideband signal. The MPRFB approach [26] shown in Fig. 2.9 consists of an analysis section and a synthesis section. By adding up the subband signals generated by the analysis section, wideband signals can be generated at the synthesis section. Thus the approach in [26] can be used for the channelization of signals of unequal bandwidths. However the bandwidths of the wideband signals generated by the synthesis section are integer multiples of the bandwidths of the subband signals generated by the analysis section. Thus the approach in [26] is not always appropriate for SDR signals, where the multiple communication standards have bandwidths which are not integer multiples of each other. Also the approach in [26] consists of a polyphase prototype filter, IDFT analysis section and DFT synthesis section. Thus the computational complexity of the MPRFB is double that of DFTFB. Also the implementation complexities associated with MPRFB have not been considered in [26].

2.5.7 The Multi-Stage Filter Bank (MSFB)

A multi-stage filter bank (MSFB) based channelizer shown in Fig. 2.10, has two stages of DFTFBs and efficient sample rate converters has been proposed in [27]. The front-end DFTFB has fixed number of channels (K_1), but the passband supports overlap with each other considerably resulting in easier isolation of channels with center frequencies of successive bandpass filters. The outputs of the front-end DFTFB are then fractionally decimated using SRCs. These decimated outputs are fed to the back-end DFTFB. Finally, the outputs of the back-end DFTFBs (K_2 channels for each back-end DFTFB) are fed to the baseband processing units as shown in Fig. 2.10. Since the sample rate is considerably lowered in the back-end, the DFTFB at the back-end needs to operate only at low speed. Due to the reduced speed requirements, the back-end DFT can be repeatedly used to extract variable bandwidth channels. The drawback of the architecture in [27] is that, since the back-end DFTFB is employed for varying bandwidth channels, hardware optimization can be done only for the fixed front-end DFTFB. The back-end needs to be changed according to the new communication standard.

Figure 2.10 Multi-stage filter bank.

2.5.8 The Cosine Modulated Filter Bank (CMFB)

An efficient architecture based on non-uniform filter banks is introduced in [28]. The non-uniform filter bank can be constructed by directly merging the relevant uniform cosine modulated filter bank (CMFB) according to the non-uniform matrix. The basic idea is that the number of channels in the uniform filter bank is firstly determined by the minimal guard bandwidth and then corresponding CMFB is designed. When the state of the input signal is known, a non-uniform matrix can be directly derived in terms of the location of every subband signal. When the state of input signal is unknown then it needs additional technique of energy detection to determine the number and location of subband signal. However the bandwidths of the wideband signals generated by the CMFB are integer multiples of the bandwidths of the subband signals. Thus the approach in [28] is not appropriate for SDR signals, where the multiple communication standards have bandwidths which are not integer multiples of each other. The computational complexity of this approach is high as it needs non-uniform multiplication and energy determination along with the original complexity of DFTFB.

2.6 Qualitative Comparison of Digital Filter Banks

A qualitative comparison of various channelizer methods is given in Table 2.2. In Table 2.2, the parameters used for comparison are computational efficiency, design flexibility which includes number of channels and possibility of independent channels, silicon efficiency and reconfigurability.

Table 2.2
Comparison of channelization approaches.

Parameters		PC approach [17]	DFTFB [17]	PFT [23,24]	TQMFB [25]	MPRFB [26]	CMFB [28]
Computational Efficiency	Multiple uniform channels	Poor	Excellent	Good	Good	Poor	Poor
	Multiple non- uniform channels	Poor	Poor	Poor	Poor	Good	Good
Silicon Efficiency		Poor	Excellent	Good	Good	Poor	Poor
Design Flexibility		Excellent	Poor	Better	Better	Good	Good
Independent Channel		Yes	No	No	No	No	No
No of channels		Selectable	2^N	2^N	2^N	Selectable	Selectable
Reconfigurability		Poor	Very Poor	Poor	Poor	Limited (the integer multiples of initial response)	Limited (the integer multiples of initial response)

The ‘computational efficiency’ is defined as the number of multiplications associated with the architecture, which includes the multiplication of filtering, modulation (DFT) and DDC to extract multiple uniform or non-uniform channels. The works in [17, 19] showed that when number of uniform bandwidth channels to be extracted is more than two, the DFTFBs outperform the PC approach. The computational complexity for extracting uniform channels using PFT, TQMFB, MPRFB and CMFB are less than PC approach, but not lower than DFTFB. This is basically because the DFTFB can utilize multi-rate method to reduce the effective working rate of the architecture and thus the complexity. But to extract non-uniform channels, DFTFB fails. The MPRFB and CMFB are designed for extracting non-uniform channels which are integer multiples of the initial prototype filter response. Hence they offer comparatively fair computational complexity for non-uniform channel extraction when compared to DFTFB. ‘Silicon efficiency’ is a measure of the area of the channelizer architecture in FPGA. It has been shown in [23] that up to 256

channels, the silicon cost of PFT approach is comparable to DFTFB, but beyond 256 channels, DFTFB outperforms PFT approach. The MPRFB and CMFB methods are based on DFTFBs, for non-uniform channel extraction. Hence they need more than twice the silicon cost of DFTFB. ‘Number of channels’ refers to the total number of channels extracted using the same architecture. The ‘number of independent channels’ indicates the total number of extracted channels which can have different bandwidth and can be uniformly or discontinuously distributed over input frequency band. Both of the above contributes to ‘design flexibility’. When design flexibility is considered PC approach supersedes all the other methods as all the channels are independent with different bandwidth and location. In PFT and TQMFB, the number of extracted channels will be in powers of two, i.e. 2^N where N is an integer (power-of-two subband stacking). The DFTFB also extracts 2^N channels, but they are uniform in nature and also has the problem of fixed channel stacking. In MPRFB and CMFB, selective number of non-uniform channels can be extracted but the selection is limited to the integer multiples of the initial prototype filter response and hence they cannot extract independent channels. On the other hand, PC approach has got excellent design flexibility but the silicon cost and computationally complexity increases linearly with the increase in the number of channels extracted.

The parameter ‘reconfigurability’ represents the adaptation of channelizer architecture to satisfy the current requirements with minimum overhead. As discussed in the previous sections, none of the existing approaches satisfy the reconfigurability requirement. The reconfigurability of MPRFB and CMFB are limited to the selection of the non-uniform channels which are integer multiples of the prototype filter response. If the prototype filter response needs to be reconfigured, the above two method fails. It can be noted from Table 2.2 that the existing channelization approaches do not offer an efficient trade-off between complexity and reconfigurability.

2.7 Summary

In this chapter, a brief review of software defined radio (SDR) is discussed and the functionalities of different blocks of SDR receiver architecture are conferred. The digital front end (DFE), especially different channelization approaches for DFE, is

given more emphasis in this chapter as it is the main area of this thesis. The channelizers in SDRs need to be implemented with low complexity and low power. Efficient implementation schemes of channelizers like per-channel (PC) approach, discrete Fourier transform filter banks (DFTFBs), Goertzel filter bank (GFB), Pipelined frequency transform (PFT), Tree structured quadrature mirror filter bank (TQMFB), Modified perfect reconstruction filter bank (MPRFB), Multi-stage filter bank (MSFB), Cosine modulated filter bank (CMFB) were also reviewed. The PC approach is the simplest approach with a distinct channelizer used for each channel, thus having a complexity directly proportional to the number of channels. Filter bank approaches like DFTFB and GFB makes use of the polyphase decomposition of filters and its complexity is independent of the number of received channels. But all the channels extracted by employing DFTFB are of uniform bandwidth. Hence for the extraction of non-uniform bandwidth channels (channels of different wireless communication standards), either distinct filter banks are required or methods like MPRFB and CMFB needs to be used. Both these methods are able to extract non-uniform channels, but are limited to integer multiples of initial prototype filter response. The complexity of MPRFB and CMFB are higher than DFTFB as it requires extra hardware for non-uniform channel extraction. But even with MPRFB and CMFB, reconfiguration of the initial response is not possible. Hence the ultimate reconfigurability with multiple independent channels, with least complexity and power, is still not possible, as required in the case of SDR. Thus there is a need for developing new filter bank architectures with reconfigurable and low complexity filters for SDR receivers. In order to accomplish the above mentioned features, the channel filter of the filter bank must be flexible and of low complexity. In the next chapter, various approaches presented in literature to minimize the hardware complexity of filters and to incorporate reconfigurability into channel filter in the filter bank of the channelizer of an SDR are discussed.

Chapter 3

Low Power and Reconfigurable Implementation Approaches for Channel Filters

Channel filtering is a process of bandpass filtering, where radio channels from a wideband input signal are extracted for further baseband processing. The channels-of-interest may be of equal or different bandwidths and may be uniformly or non-uniformly, continuously or non-continuously distributed over the input frequency band. Thus, the filters have to attenuate adjacent-channel interferers and must meet the blocking characteristics specified by the relevant communication standard. Due to these stringent adjacent channel specifications, higher-order channel filters are required in the receiver. As the order of the filter increases, the complexity of the filter also increases, which in-turn increases the area and power consumed. Therefore, implementing digital filters with small area and low power consumption is vital for reducing the overall complexity of the channelizer. The channel filters are typically placed directly after the ADC. Hence they have to operate at high sampling rates and consequently high-speed. Thus channel filter having low power and high-speed characteristics need to be investigated. In this chapter, a review of different approaches for low power and low complexity implementation of channel filters has been presented. Flexibility or reconfigurability is another key parameter to be considered in software radio channelizers. This chapter also presents an account of different reconfigurable filter architectures in literature.

3.1 Channel Filters for Wideband Receivers

Digital filters can be Infinite Impulse Response (IIR) filters or Finite Impulse Response (FIR) filters. IIR filters can be realized using recursive algorithms as it is an infinite length impulse response filter. The main advantage of IIR filters is that they can produce a steeper slope for a given number of coefficients when compared to FIR filter. But IIR filters are difficult to be reconfigured due to the presence of poles. This is because IIR filters can become unstable due to the new pole values after reconfiguration. But no such problems exist for FIR filters due to the absence of poles.

The main advantage of FIR filters is that the group delay is constant. This provides the capability of obtaining both a steep cutoff and linear phase response. To achieve a given filter response characteristic, FIR filters have the disadvantage that they require more area and cost than its IIR counterpart. The need of higher-order filters increases the cost of FIR filters when compared to IIR filters. FIR filter is an all-zero filter and hence it is absolutely stable. Compared to IIR filters, FIR filters offer the following advantages:

- FIR filters are linear phase filters. They delay the input signal, but don't distort its phase.
- FIR filters are suited to multi-rate applications. Whether the signal is decimated or interpolated, the use of FIR filters allows omission of some calculations, thus providing an advanced computational efficiency. In contrast, if IIR filters are used, each output must be individually calculated, even if that output is discarded (so the feedback will be incorporated into the filter).
- In practice, all digital filters must be implemented using *finite-precision* arithmetic, that is, a specific number of bits. The use of finite-precision arithmetic in IIR filters can cause stability problems due to the use of feedback. But FIR filters have no feedback, so they can usually be implemented using fewer bits, and the designer has fewer practical problems to solve related to non-ideal arithmetic.
- FIR filters are less sensitive towards coefficient quantization compared to IIR filters.

FIR filters are widely used as channel filters due to their linear phase property and absolute stability. FIR filters can be realized by direct form (DF) or transposed direct form (TDF). The direct form realization, which is the most intuitive implementation of the time domain filter transfer function, is shown in Fig. 3.1. The FIR filter equation is

$$y[n] = \sum_{k=0}^{N-1} x[k]h[n-k] \quad (3.1)$$

where $x[k]$ is the k -th sample of signal, $y[n]$ is the output for the n^{th} sample, N is the total number of taps (coefficients) of the filter and h_k is the coefficient value for the k^{th} tap.

Figure 3.1 Direct form FIR filter structure.

A DF FIR filter output is formed by accumulating the result of delayed versions of input signal, multiplied by the filter tap coefficients. The critical path delay of the DF filter is given by:

$$T_{df} = t_{mult} + (N - 1)t_{add} \quad (3.2)$$

where t_{mult} is the multiplier tap delay and t_{add} is the delay of the adder. Thus the critical path length for DF FIR filter is directly proportional to the number of filter taps, N . For the direct form N -tap FIR filter shown in Fig. 3.1, N multiplications and $(N-1)$ additions are needed. As a result, the speed of operation of the DF FIR filter is relatively low especially when higher-order channel filters are required. The filter length is determined by the transition bandwidth and stopband attenuation specifications of the filter. A narrow transition band filter would require more number of filter taps. The critical path delay of DF structure can be reduced by applying transposition theorem to obtain the TDF FIR filter as shown in Fig. 3.2(a) and equation (3.3).

$$y[n] = \sum_{k=0}^{N-1} h[k]x[n-k] \quad (3.3)$$

The critical path delay of the TDF FIR filter is given by:

$$T_{tdf} = t_{mult} + t_{add} \quad (3.4)$$

Since the critical path delay is independent of the number of taps, TDF has been the preferred architecture for high-speed, higher-order FIR channel filter implementations in wireless communication receivers. For linear phase N -tap FIR filter, as the filter coefficients are either symmetric or anti-symmetric, the number of multiplications can be reduced from N to $N/2$ for N even and to $(N + 1)/2$ for N odd, which is 50% reduction compared to DF structure. Thus the TDF structure offers the flexibility of high-speed operation at low power compared to the DF structure. It can be noted that the coefficient multiplication in TDF FIR filter can be done in parallel. This parallel multiplication block represented as ‘multiplier block’ is shown in Fig. 3.2(b). The multiplication is the most complex operation in digital filtering. The main advantage of the multiplier block is that the parallel multiplication can be optimized by substituting multiplications with shifts and additions. Hence this research work focuses mainly on the optimization of TDF structure. In the following sections, different FIR filter realization approaches along with their complexity and reconfigurability issues are reviewed.

Figure 3.2 (a) Transposed direct form FIR filter structure, (b) Modified TDF FIR filter structure with a common multiplier block.

3.2 Computational Complexity Analysis of Channel Filters

The basic computations involved in digital filtering are multiply and accumulate (MAC). The accumulate operation is done by adders. The complexity of filter implementation is dominated by the number of multiplications as multiplication requires multiple adders and shifts for implementation. If dedicated multipliers are used for each coefficient multiplication in the channel filters of a filter bank channelizer, it will result in a large chip area and huge power consumption. This is unacceptable from a hardware complexity point of view. Digital filters employed in battery powered mobile systems must be realized to consume less power and operate at high-speed. In the work presented in this thesis, only TDF filters, whose critical path delay is independent of the filter length, are used on account of its high-speed of filtering compared to DF filters. Although programmable filters based on DSP cores offer the advantage of flexibility, they are not suitable for mobile applications that demand high throughput and low power consumption.

In a digital filter, the most power consuming operation is coefficient multiplication, which is implemented using shifts and additions. The number of adders (subtractors) needed to implement the coefficient multiplication operation dominates the complexity of the fixed coefficient digital filters as shift operations can be hardwired. The adders termed as Logic Operators (LOs) are required for computing the sum of partial products in the coefficient multiplier. The number of LOs is significant in determining the area and power consumption requirements of FIR filters. The speed of the filter circuit is determined by the number of adder-steps in the maximal path of decomposed multiplications termed as Logic Depth (LD). Hence, the methods that focus on low power and low complexity FIR filter implementations target on reducing the two metrics, LOs and LD, in coefficient multipliers. The efficiency in reducing the cost metric (power, area, and LD) of the filter directly influences the performance of target hardware.

The number of adders for realizing coefficient multiplication is directly proportional to the number of nonzero digits (bits) present in the quantized filter coefficients. Thus number systems with fewer number of nonzero digits are widely used for the representation of filter coefficients compared to conventional binary

system [29-31]. Among such number representations, the canonical signed digit (CSD) [30] representation and minimal signed digit (MSD) are the most popular ones. The CSD number system is a signed digit number system that minimizes the number of nonzero digits and thus reduces the number of partial product additions in a hardware multiplier. The encoding scheme uses a digit set that is ternary and each digit can be either -1, 0, or +1. In CSD, no two adjacent digits are nonzero, i.e. A number $c_0c_1c_2\dots c_{n-1}$ is said to be in CSD representation if $c_i * c_{i+1} = 0$. This property implies that for an arbitrary n -bit 2's complement number (binary representation), there are at most $\left\lfloor \frac{n}{2} \right\rfloor$ nonzero digits. For a 2's complement number, there can be n nonzero digits for an n -bit number. On an average, CSD representation offers a reduction of 33% of nonzero digits compared to the binary representation [30]. Therefore it is possible to implement a coefficient multiplier with 33% less number of adders in CSD form compared to its binary counterpart [30].

In [32], a minimum signed digit (MSD) representation was proposed which resulted in the least number of nonzero digits. But the main drawback of the MSD representation is that, it can have more than one representation format for the same decimal or binary representation. The main problem with these alternative number systems such as CSD and MSD compared to binary representation is that, they involve subtraction operation which will have reconfiguration constraints if the filters realized using these number representations need to be reconfigured. Also if dynamic reconfigurability is required as in SDR channel filters, on the fly conversion of binary filter coefficient values into CSD or MSD would incur conversion overhead. The conversion overhead could impose constraints on real time reconfigurability. In addition to this, there is additional memory requirement for storing the signed bits in case of CSD or MSD representations compared to binary representation.

Once the filter coefficient is represented in the chosen number system, the next option will be to look for further hardware optimization by minimizing the number of nonzero bits while satisfying the filter characteristics such as passband ripple and stopband attenuations [33-35]. In [33], the filter coefficients are first scaled according to a local search before quantization process is performed resulting in a filter with much better frequency response characteristics compared to the original sum of power

of two terms. In [34], the coefficient optimization is done as a polynomial optimization approach. The filter coefficients are scaled and normalized and then multiplied by a variable scale factor. The algorithm in [34] continues in an iterative way and takes up lot of computation time. In [35], a hybrid scheme is formed by integrating the main features of an adaptive genetic algorithm (AGA), simulated annealing (SA) and tabu search algorithm. AGA with varying population size of the quantized filter coefficients and varying probabilities of genetic operations is used as the basis of the hybrid algorithm. SA becomes significant when AGA gets stuck in a local optimum. The use of SA is to help AGA escape from local optima and prevent premature convergence. The concept of tabu is introduced to increase convergence speed by reducing search space according to the properties of filters coefficients. But the method is only appropriate for short wordlength coefficients. In general, the coefficient quantization or optimization approaches [33-35] try to modify the filter coefficients in view of minimizing the number of nonzero bits for reducing the number of adders required for partial product addition. But the modifications of coefficient values alter the frequency response characteristics and hence above methods are generally lossy approaches.

Digital filtering can be treated as multiplication of one variable (input signal) with multiple constants (coefficients) named as multiple constant multiplications (MCM) [36]. MCM is a transformation closely related to the widely used substitution of multiplications with constants by shifts and additions. While the latter considers multiplication with only one constant at a time, MCM considers simultaneous multiplications of multiple constants. The advantage of MCM is that it provided the chance for eliminating repetitive or redundant multiplications. This change in the scope of the transformation significantly enhances its power and domain of application. In [36], the MCM idea was extended to application-specific FIR filters, by considering input as a variable and the filter coefficients as multiple constants. Since the coefficients of application-specific filters are fixed and are considered as constants, they can be analyzed for repetitive additions (subtractions) using identical bit patterns present in the CSD representation of coefficients called common subexpressions (CSs). A method to eliminate the most commonly occurring CSs such as $[1\ 0\ 1]$ and $[1\ 0\ -1]$ in the CSD representation of filter coefficients was proposed in [37]. As an additional criterion in the subexpression identification process, an estimation of a latch count improvement was also considered in [37]. The timing and

routing delay has been investigated in [37] when more and more CSs are employed for adder reduction. It explicitly considers the number of delay latches in the circuit and attempts to minimize a combination of the number of adders and the number of delays. The CSs across the adjacent coefficients are also considered in [37]. But as the horizontal subexpressions $[1\ 0\ 1]$ and $[1\ 0\ -1]$ are more frequently occurring, they are used for obtaining results. The simulation results in [37] showed that the number of adders could be reduced by 50% over conventional CSD implementations.

An MCM method [38] was proposed for FIR filters which were applicable to both DF and TDF structures to exploit the common computations between multiplications. It was shown in [38] that when the filter coefficients are represented in CSD, certain 3-bit CSs like $[1\ 0\ 1]$ and $[1\ 0\ -1]$ and their negated versions are repeated many times. These repetitive patterns (CSs) can result in redundant additions and hence they need to be implemented only once. Thus in [38], an iterative algorithm was proposed to eliminate these redundant patterns by forming CSs only once. Thus by employing the common subexpression elimination (CSE) technique, it has been shown that 38% reduction in the number of adders is achieved in MCM based FIR filter structures [38].

A modification of the common subexpression optimization technique in [38] is proposed in [39] for identifying proper patterns for elimination so that the optimization impact is maximized. It is based on the CSE technique and combines the exhaustive search for multiple pattern identification with a steepest descent approach for pattern selection. The result shows a significant reduction in either arithmetic operations or hardware necessary to implement those operations combined with satisfactory runtimes.

The CSE techniques were applied to MSD representation of filter coefficients in [32]. The MSD representation is attractive because it provides various forms that have the minimal number of nonzero digits for a constant. This redundancy can lead to efficient filters if a proper MSD representation is selected for each constant. But the proper selection of representation has no general rules. Hence if the selection is not proper, it can be a drawback. Starting from the CSD representation, all the MSD representations are discovered by repeatedly applying simple transforms,

$[1\ 0\ -1] \rightarrow [0\ 1\ 1]$ and $[-1\ 0\ 1] \rightarrow [0\ -1\ -1]$. The proposed filter synthesis algorithm in [32] is to select one coefficient at a time and synthesize it using previously synthesized patterns, which is different from the conventional method that searches subexpressions common to multiple constants. In [32], there was no consideration for logic depth as the whole emphasis was for reduction of LOs. Also in [32], MSD representation is obtained from CSD representation of filter coefficients by some re-ordering. Hence there is a possibility of change in the frequency response characteristics of the filter.

The method in [37] was modified in [40] by developing a non-recursive signed CSE (NR-SCSE) algorithm that minimizes the LD of the filter structure. The new algorithm has been called NR-SCSE because it searches for the non-recursive signed common subexpressions that must be eliminated from the original CSD array. While the recursive utilization of a common subexpression generates a high logic depth, the NR-SCSE allows the designer to overcome the problem of high LD by using each subexpression only once. NR-SCSE algorithm [40] gives two options to select the order of subexpressions. The first option allows us to give a higher-order for the subexpressions, which will result in fewer numbers of LOs at the cost of an increased LD. The second option provides the user to select a lower order for subexpressions, which will give the best reduction of LD at the cost of a slight increase in the number of LOs. As the order of the subexpressions increase, the number of full adders for realizing each subexpression will also increase. NR-SCSE offers the good relation with respect to number of adders and LD along with optimal runtime operation, due to its property that the common subexpressions belonging to different coefficients are not shared.

A novel CSE method was proposed in [41] for the implementation of FIR filters with a good trade-off between LOs and LD. The method in [41] can be regarded as the first approach for a CSE algorithm resulting in minimum adders with a reasonably shorter LD. The method can compute the results for a higher-order filter with out taking much time. Authors claim that, this was the first CSE method that can perform trade-off designs between complexity and the throughput rate.

Method in [42] studies the design and multiplier-less realization of a new SDR receiver with reduced system delay. It employs low delay FIR and digital allpass

filters to effectively reduce the system delay of the multi-stage decimators in the receivers. The optimal least-square and minimax designs of these low-delay FIR and allpass-based filters are formulated as a semidefinite programming (SDP) problem. Design results show that the proposed architecture considerably reduces the system delay and implementation complexities (especially in high-speed variable multipliers) as compared with conventional approaches.

A new algorithm for the problem of multiplication by constant was presented in [43]. The algorithm in [43] is based on extensions and enhancements of previous algorithms from [39]. They have generalized the previous results by dealing with the problem of the optimization of multiplication of one vector by one constant matrix. The proposed method, based on number recoding and dedicated common subexpression factorization algorithms, was implemented in a VHDL generator. Compared to the best previous results, solution of [43] leads to a significant reduction in the total number of additions, up to 40 percent.

In general, the methods in [36-43] utilize the most CSs within a coefficient called as the Horizontal CSs (HCSs) and so, those methods can be referred to as horizontal common subexpression elimination (HCSE). HCSE method is illustrated below using the CSD representation of filter coefficient $h_k = [0.10-1010101010010-1]$. In direct implementation using shift and add, the output of the filter is

$$h_k = 2^{-1}x_1 - 2^{-3}x_1 + 2^{-5}x_1 + 2^{-7}x_1 + 2^{-9}x_1 + 2^{-11}x_1 + 2^{-14}x_1 - 2^{-16}x_1 \quad (3.5)$$

where x_1 is the input signal. It requires 7 adders/subtractors to implement (3.5). It can be noted that the bit patterns $[1 \ 0 \ 1]$ and $[1 \ 0 \ -1]$ are repeated twice in h_k , which can be expressed as CSs $x_2 = x_1 + 2^{-2}x_1$ and $x_3 = x_1 - 2^{-2}x_1$. Using CSs, filter output can be expressed as

$$h_k = 2^{-1}x_3 + 2^{-5}x_2 + 2^{-9}x_2 + 2^{-14}x_3 \quad (3.6)$$

Here we need only 5 adders (2 for CSs and 3 for equation (3.6)) for HCSE implementation, which results in saving of 2 adders from direct CSD implementation.

A vertical common subexpression elimination (VCSE) technique has been proposed in [44], where the authors exploit the fact that many vertical common subexpressions (VCSs) exist between adjacent filter coefficients. VCSs are those CSs that occur between adjacent coefficients. The structure proposed in [44] is, however, the more efficient than the structure obtained using HCSE only in the case where coefficient wordlength is relatively small.

In [45], CSE techniques for CSD were applied by combining four types of subexpression elimination techniques: (1) HCSE (2) VCSE (3) Horizontal Super Subexpression Elimination (HSSE), which eliminates redundant horizontal common subexpressions with identical shifts between them, and (4) Vertical Super Subexpression Elimination (VSSE), which eliminates redundant vertical common subexpressions with identical shifts between them. The design examples of channelizers of various cellular standards presented in [45], show that the optimization techniques offer an average LO reduction rate of 50% over conventional channel filter implementations.

A comparison of hardware reductions achieved using the HCS and VCS in realizing digital filters is done in [46]. The authors show that the CSE technique employing horizontal CS offer better reductions of LOs and LD than their vertical CS counterpart in practical FIR filter implementations. The simulation results show that the hardware reductions offered by the vertical CS are improved in implementing IIR filters than their FIR counterparts. The CSD-based CSE methods in [36]-[45] suffer from the drawback that the symmetry of FIR filter coefficients cannot be completely exploited when the bits in VCSs are of opposite sign. As a result, additional LOs are required to obtain the symmetric part of the coefficients when more than one VCSs with bits of opposite sign exist [46].

The VCSE technique can be explained using the example in Table 3.1. The numbers in the first row represent the number of bitwise right shifts. The coefficient multipliers are realized using the VCSs $[1 \ 1] = x_4 = x_1 + x_1[-1]$ and $[1 \ -1] = x_5 = x_1 - x_1[-1]$, where $x_1[-k]$ represents input x_1 delayed by k units. Using VCSs, the filter tap output can be expressed as

$$y_k = 2^{-1}x_5[-k] - 2^{-3}x_5[-k] + 2^{-5}x_4[-k] - 2^{-8}x_1[-k] \quad (3.7)$$

The output for the symmetric part of (3.7) is given by

$$y_{n-k-1} = -2^{-1}x_5[n-k-1] + 2^{-3}x_5[n-k-1] + 2^{-5}x_4[n-k-1] - 2^{-8}x_1[n-k-2] \quad (3.8)$$

where n represents the filter length.

Table 3.1
FIR filter coefficients.

Bit shift(BS)	1	2	3	4	5	6	7	8
h_0	1	0	-1	0	1	0	0	-1
h_1	-1	0	1	0	1	0	0	0

It has been reported in [46] that y_{n-k-1} cannot be obtained from y_k by a simple delay operation (delaying y_k by $(n-k-1)$) due to the sign differences of the first and second terms in (3.7) with corresponding terms in (3.8). Note that there exists another difficulty due to the delay differences between the first and second terms of (3.8) (delay = $n-k-1$) with that of the last term (delay = $n-k-2$). In this case, an extra LO would be required to compute the last term of (3.8). Thus to conclude, when VCSE is used, additional LOs are needed not only when the bits are of opposite signs exist as pointed out in [46] but also when the subexpressions and/or unpaired bits which exist in the symmetric second half coefficients have different delays compared to the symmetric first half coefficients. This requirement of extra adders poses constraints in reducing the number of LOs adders when CSD based VCSE is used.

A new binary based CSE method was introduced in [47] called Binary Subexpression Elimination (BSE) method was proposed, which produced better reduction of LOs compared to the CSD-based CSE methods [36]-[46]. The reduction of LOs was obtained in [47] by combining three techniques - the 3-bit binary HCSE, HSSE and VCSE. The BSE method [47] offers the best reduction of adders with negligible increase in LD when compared with existing CSD-based CSE methods, especially for higher-order FIR filters. As BSE method uses binary representation of coefficients which does not have negative bits, the overhead of additional LOs inherent in CSD-based VCSE methods (due to negative bits in CSD representation) does not exist in BSE. However, the overhead of additional LOs occur not only due to

the presence of bits of opposite signs in the coefficient representation, but also due to the delay differences between nonzero terms (VCSs and unpaired nonzero bits) of adjacent coefficients, when VCSE is applied. The detail explanation is presented in Section 4.1 of Chapter 4. Though the BSE method [47] does not have the former overhead, it suffers from the latter overhead.

A different approach of graph theory is adopted for LO reduction in [48-52]. In these works, the whole multiplier block is synthesized using similar graph synthesis algorithms and hence called as Graph dependent (GD) algorithms. The primary focus of GD algorithms based on graph theory is on reducing the number of LOs, with less emphasis given to LDs. The Bull-Horrocks (BH) algorithm [48] provides the notion of primitive operator by applying the graph theoretic approach to systematically minimize the LO cost of the multiplication block of the transposed direct-form FIR filter. In GD algorithms, the fundamentals are generated one at a time depending on the previously generated fundamentals. The BH algorithm used only adders for generating fundamentals and this created in high values of new fundamentals skipping the intermediate values.

The BH algorithm was modified in [49] as Bull-Horrocks Modified (BHM) algorithm. A new algorithm called n -dimensional Reduced Adder Graph (RAG- n) algorithm was also proposed in [49]. Both BHM and RAG- n made use of the subtractors which resulted in low valued fundamentals. The main difference between BHM and RAG- n algorithm is that the coefficient which requires the least number of adders is synthesized first in the latter method, whereas the coefficients are synthesized in the previously defined order in the former algorithm. Although the multiplier block from the RAG- n algorithm is superior to that from BHM, the former method requires large memory and much computation time. The GD algorithms are good for minimizing the number of LOs (and hence reducing the area) but are not optimal for speed. Generally GD algorithms synthesize coefficients sequentially, and thus greedily reduce the hardware complexity with very little regard to the adjacent coefficients. This in-turn results in inadequate exploitation of redundancies among coefficients. Both [48] and [49] produced multipliers with longer LDs which will increase the delay.

A modification of GD algorithms in [48] is proposed in [50], for a good trade-off between LOs and LD. Three new delay reduction methods such as tree reduction, limited selection method and minimum adder step method are also reviewed in [50]. These three reduction methods are combined with BHM and RAG-n to form two new algorithms which offered better trade-off between LOs and LDs. But the inherent drawbacks of BHM and RAG-n of inadequate exploitation of redundancies among coefficients are not addressed in [50].

A Contention Resolution Algorithm (CRA) was proposed in [51] which also produced multipliers with fewer LOs. The CRA algorithm is based on CSD-based CSE method in which graphs are used for solving the CSE. It manages 2-bit CSs and aims at achieving the minimal LD as the primary goal. But the method resulted in an increase in the number of LOs in many cases.

A new GD algorithm was proposed in [52] to reduce the number of LOs. The main idea behind the method in [52] is to use a better heuristic approach to synthesize intermediate fundamentals to jointly optimize for all target constants. But the method in [52] is computationally more expensive than [49] as it explores a very large space of possible intermediate fundamentals. The method in [52] resulted in longer LD.

In general, as GD algorithms [48-52] synthesize each coefficient sequentially, they suffer from two drawbacks - increased computation time for optimization and longer LD which makes them less attractive for high-speed filter. Though the increased computation time may not be a constraint for fixed-coefficient filters (as the coefficients can be synthesized beforehand), the increased LD will have adverse effects on the speed of filtering operation. Therefore, the CSE methods offer a high-speed solution to higher-order filtering compared to GD methods. Therefore the work presented in this thesis is based on CSE methods on account of their low power and low complexity implementations.

3.3 Filter Reconfigurability

Several implementation approaches for reconfigurable FIR filters have been proposed in literature [53-60]. These designs include either a fully programmable Multiply-Accumulate (MAC) based filter processor or dedicated architectures where the filter coefficients can be stored in registers. The architecture of a filter processor consists of a datapath with a single MAC unit, data and program memories, and a control unit is discussed in [53]. The datapath includes a 16-bit adder (subtractor), a multiplier and a 32-bit accumulator. The performance of the processor is mainly restricted by the delay of this datapath, more specifically that of the multiplier. The main disadvantage of the filter processors is that the area and power requirements are significantly large.

A programmable digital FIR filter for high-performance and low power applications is proposed in [54]. The architecture is based on a computation sharing multiplier (CSHM) which specifically targets computation re-use in vector-scalar products and can be effectively used in the low complexity programmable FIR filter design. Efficient circuit-level techniques, namely a new carry-select adder and conditional capture flip-flop (CCFF), are also used to further improve power and performance.

In [55], a CSD-based digit reconfigurable FIR filter architecture was proposed. This architecture is independent of the number of taps because the number of taps and nonzero digits in each tap are arbitrarily assigned. The intention of the authors was to reduce the wordlength of the coefficients and thus the filter complexity without affecting the filter performance. But the architecture in [55] demanded huge hardware resources and this makes the method infeasible for resource constrained SDRs. Also the architecture in [55] is digit based and hence the speed of operation is low.

In [56], a high-speed and programmable CSD-based FIR filter was proposed. The filter architecture consisted of a programmable CSD-based Booth encoding scheme and partial product Wallace adder tree. The final adder was a carry look ahead adder. This method resulted in high-speed but at the cost of high power consumption.

The aim in [56] was solely to design a high-speed reconfigurable filter and no consideration was given for reducing power consumption.

In [57], the concept of reconfigurable multiplier block (ReMB) was used. The ReMB consisted of a coefficient store and a general-purpose multiplier. The ReMB will generate all the coefficient products and a multiplexer will select the required ones depending on the input. It was shown that, by pushing the multiplexer deep into the multiplier block architecture, the redundancy can be reduced. The resulting specialized multiplier design is more efficient in terms of area and computational complexity compared to the general-purpose multiplier plus the coefficient store [57]. But the ReMB proposed in [57] has its area, power and speed dependent on the filter-length making them inappropriate for higher-order FIR filters.

In [58], a time multiplexed MCM approach was proposed. This method considers the coefficient set as constant and uses the GD algorithms for reducing redundancy. But this method follows a directed acyclic graph (DAG) structure which will result in long LD and thus lower speed of operation. Also the area of the architecture linearly increases with the filter length as in [57] and filters with filter-length above 40 are infeasible.

An area-efficient programmable FIR digital filter using CSD coefficients was implemented in [59] that use a switchable unit-delay to allocate the desired number of nonzero CSD coefficient digits to each filter tap. The area efficiency of this method highly depends on the number of nonzero coefficients which is fixed by the filters stopband and passband specification. Here no efforts has been made to reduce the complexity of the multiplier apart from using CSD coefficients and hence will result in larger delay and power.

In [60], a reconfigurable FIR filter design is presented using dynamic partial reconfiguration (DPR) is presented. DPR allows a part of the device to be modified while the rest of the system continues to operate, unaffected due to reprogramming. The SDR needs very sharp transition bandwidth filter with a very high order, thus resulting in large number of slices for FPGA for the method in [60].

The architectures in [54-60] are appropriate only for lower-order filters and hence not suitable for channel filters in SDR receivers, which are normally higher-order filters due to stringent adjacent channel attenuation specifications. From the review studies, it was found that, integration of low power, low complexity and reconfigurability into hardware architecture for FIR filters and filter banks have been hardly addressed in literature.

3.4 Summary

In this chapter, various low power, low complexity and reconfigurable implementations of filters have been reviewed. The complexity of filtering operation is dominated by the coefficient multiplication operation. It was found that CSE techniques [36-47] and GD algorithms [48-52] have been best suited for reducing coefficient multiplier complexity since they consider coefficient multiplication problem as a MCM problem with input signal as variable and coefficients as constants. GD algorithms resulted in better reduction of adder complexity in coefficient multipliers at the cost of increased delay compared to CSE techniques. As the channel filters in SDR receivers come directly after ADC, they need to operate at high-speed. Therefore CSE techniques are more appropriate for high-speed channel filters in SDR receivers than GD algorithms. The channel filters also need to be reconfigured for multi-standard operation as in the case of SDR. More research has to be done for the efficient use of CSE techniques for the higher-order reconfigurable channel filters in SDR receivers. In the next chapter a new binary based CSE technique is proposed for implementing higher-order low power FIR filters.

Chapter 4

Low Power FIR Channel Filter Realization Using Binary Common Subexpression Elimination

The channel filters employed in mobile SDR are generally higher-order filters, and must be realized to consume less power and operate at high-speed. Although programmable filters based on digital signal processor cores are available, they are not very efficient as they consume more power and operate at low-speed. Hence dedicated FIR filter architectures have received great deal of attention in the last decade. The number of adders (subtractors) used to implement the coefficient multipliers determine the complexity of the FIR filters. A new 4-bit binary representation-based common subexpression elimination (BCSE-4) method which employs 4-bit common subexpressions (CSs) for implementing higher-order low power FIR filters with fewer number of adders than CSD and binary based CSE methods is presented in this chapter. The proposed algorithm reduces the complexity of coefficient multipliers by combining two techniques: Binary Horizontal Subexpression Elimination (BHSE) and Binary Horizontal Super Subexpression Elimination (BHSSE). The reduction of adders is achieved with at the most 1 adder-step increase in critical path length of filter coefficient multipliers. The proposed method offers a good trade-off between adders and logic depth (critical path length) in comparison with methods in [37-52] especially for higher-order filters.

4.1 Binary Common Subexpression Elimination Algorithm

In this section, a new CSE algorithm based on binary representation of filter coefficients is proposed. The proposed technique is called 4-bit Binary Common Subexpression Elimination algorithm (BCSE-4), which completely exploits the symmetry of the FIR filter coefficients. The BCSE-4 consists of two steps. First the redundant subexpressions within each coefficient known as Binary Horizontal Common Subexpressions (BHCSs) are identified and eliminated using BHSE algorithm. Subsequently, the BHSSE algorithm

is used to eliminate redundant identical shifts between BHCSs. The number of adders (subtractors) needed to implement the coefficient multiplier is directly proportional to the number of nonzero bits present in the coefficient. Among different coefficient representations, CSD representation employed in CSE techniques [37-46] is the most popular number representation due to the fact that it has 33% fewer nonzero bits compared to its binary counterpart. But the overhead of additional LOs in filters realized using VCSE method due to the presence of bits of opposite signs in the CSD representation would cancel out the inherent advantage of fewer nonzero bits in CSD as explained in Section 3.2 of Chapter 3. On the other hand, as the binary representation does not have negative bits, it has two advantages over CSD in realizing FIR filters. First, there is no overhead of additional LOs due to sign change as Binary-VCSE uses only positive numbers. Second, coefficient multipliers can be realized without using subtractors (solely using adders) and hence incorporating reconfigurability into the filter would be easier when coefficients are coded using binary. If dynamic reconfigurability is needed as in the case of adaptive filters and filters employed in multi-standard wireless communication receivers, then on-the-fly conversion of binary values of filter coefficients to CSD values would incur conversion overhead. As the number of nonzero digits turns to be minimum, CSD-based CSE has less initial optimization space whereas in binary there is more space for optimization. However, it should be noted that more optimization space does not necessarily guarantee fewer LOs. Using statistical analysis, it has been shown in [47] that the number of LOs, N_{LO} , can be related to the number of nonzero bits, N_{nz} , the number of CSs, N_{cs} , and the number of unpaired bits after common subexpression grouping, N_{up} , using the expression:

$$N_{LO} \approx (0.2345 \times N_{nz} - 0.6643 \times N_{cs} + 4.0487 \times N_{up}) \quad (4.1)$$

Based on statistical analysis of various FIR filters corresponding to different frequency response specifications, filter lengths and coefficient wordlengths, it has been shown in [47] that the value of N_{nz} in binary representation is only 2 times more than that in CSD representation in the worst case. On an average, N_{cs} is found to be 6.5% higher for binary, which is only a slight overhead over CSD. It can be noted from (4.1) that, the weight for N_{up} is substantially larger compared to the weights of N_{nz} [17.3 times] and N_{cs}

(6.1 times). Therefore, the number of LOs is largely dependent on N_{up} . The average reduction of N_{up} for binary filter coefficients over CSD values is found to be 67%. The substantial reduction of N_{up} which has a significant weight over N_{nz} and N_{cs} , outperforms the slight increments of N_{nz} and N_{cs} . Consequently, binary representation of coefficients would produce better reduction of LOs for FIR filters. This is true especially for higher-order FIR filters because the reduction of N_{up} is even larger as the filter length increases. Also, the possibility of finding a binary common subexpression is 2.73 times more than a CSD common subexpression for same wordlengths [47]. This would have an adverse effect on maximizing the formation of CSs for CSD-based CSE, which in turn would restrict the elimination of maximum number of CSs. The above explanation clearly indicates that even though binary representation has more number of nonzero bits, the number of CSs can be eliminated is substantially larger than that of CSD and hence only fewer number of LOs is needed for realizing the multiplier blocks of FIR filters. It can also be noted that, the superiority of the binary representation solutions increases as the number of taps and bit-width of the constants (coefficients) increase [82]. Hence for a higher-order filter, binary based representation will have a greater advantage over MSD and CSD representation by forming more CSs which in turn helps in maximizing the elimination of number of CSs.

A 3-bit binary based CSE (BSE) is proposed in [47] is the best known CSE technique. In this thesis, the BSE method refers to the one in [47]. The motivation to use 4-bit binary CSs in the work in this chapter is the possibility to reduce the number of LOs without affecting the LD when compared to the BSE method which uses 3-bit binary CSs. The term ‘4-bit subexpressions’ means that the maximum number of nonzero bits in the subexpression is 4. Thus, subexpressions which have 2, 3 and 4 ones are considered in the proposed CSE algorithm. The proposed BCSE-4 differs from the BSE in two aspects. First, BCSE-4 uses 4-bit CSs (CSs that have 2, 3 and 4 ones) whereas the BSE uses 3-bit CSs (CSs that have 2 and 3 ones). As more number of nonzero bits are grouped in BCSE-4, greater LO reduction is possible compared to BSE. The second difference is that the proposed BCSE-4 excludes the use of VCSs in order to eliminate the overhead of additional LOs due to the delay differences between the VCSs, whereas the BSE did not

account for this overhead. This can be explained using the example of an 8-tap, 12-bit wordlength FIR filter shown in Table 4.1.

The numerals in the first row of Table 4.1 represent the number of bitwise right shifts. If N_z is the number of nonzero bits in the symmetric first half coefficients and N is the filter length, the number of LOs (which includes the adders for coefficient multiplication and the inter-tap adders) needed to implement the filter using direct implementation is $(N_z - 1) + \lceil N/2 \rceil$ [45], where $\lceil x \rceil$ represents the largest integer less than x .

Table 4.1
Binary representation of 8-tap FIR filter coefficients.

Number of bitwise right shifts →	1	2	3	4	5	6	7	8	9	10	11	12
Coefficients ↓												
h_0	1	0	0	0	0	0	1	1	0	1	0	0
h_1	1	0	0	0	1	0	1	0	1	1	0	1
h_2	0	0	0	1	0	0	0	1	0	1	1	1
h_3	0	0	0	1	0	0	0	1	1	1	0	0
$h_4=h_3$	0	0	0	1	0	0	0	1	1	1	0	0
$h_5=h_2$	0	0	0	1	0	0	0	1	0	1	1	1
$h_6=h_1$	1	0	0	0	1	0	1	0	1	1	0	1
$h_7=h_0$	1	0	0	0	0	0	1	1	0	1	0	0

The direct implementation is defined as the implementation of the multipliers using shifts and adds, without employing any CSs. In this example, N_z is 19 and N is 8. Thus 22 LOs would be required to implement the filter. While reducing the number of LOs, it is important to ensure that it has minimal impact on the LD of the multiplier. It has been shown in [45] that the binary tree-structured partial product addition scheme produces multipliers with low LDs on account of parallel operation. In the proposed method, binary tree structure is used to ensure low LD. The LD of an n -bit CS for the binary tree-structured adder is $\lceil \log_2 n \rceil$ adder steps. For direct implementation, the LD=3

adder-steps for the example in Table 4.1. When HCSE is employed, the Horizontal CSs (HCSs) $x_2 = [1 \ 1]$, $x_3 = [1 \ 0 \ 1]$, and $x_4 = [1 \ 1 \ 1]$ are indicated inside bold rectangles. If x_1 represents the input signal, using above HCSs, the output y_1 corresponding to coefficients h_0 to h_3 can be written as

$$y_1 = 2^{-1}x_1 + 2^{-7}x_2 + 2^{-10}x_1 + 2^{-1}x_1[-1] + 2^{-5}x_3[-1] + 2^{-9}x_2[-1] + 2^{-12}x_1[-1] + 2^{-4}x_1[-2] + 2^{-8}x_3[-2] + 2^{-11}x_2[-2] + 2^{-4}x_1[-3] + 2^{-8}x_4[-3] \quad (4.2)$$

Fig. 4.1 shows the filter realization. The numerals shown in the schematic represents the right shift by the respective numeral value. 18 LOs with an LD of 3 are required to realize the example filter in Table 4.1 using HCSE (11 LOs for (4.2), 3 LOs for the HCSs x_2 , x_3 and x_4 , and 4 inter-tap LOs for the symmetric second half coefficients).

Figure 4.1 Filter realization using horizontal CSE.

Note that although x_4 has 3 ones, only one adder is needed to realize x_4 as the output of x_2 is shared while realizing x_4 . If VCSs are employed as in BSE method, the VCS, $[1 \ 1]$, given by $x_5 = x_1 + x_1[-1]$, is indicated inside dotted rectangles in Table 4.1. Note that $x_1[-k]$ represents the input x_1 delayed by k units. Using VCSs, the output y_2

corresponding to coefficients h_0, h_1, h_2 and h_3 can be expressed as

$$y_2 = 2^{-1}x_5 + 2^{-7}x_2 + 2^{-10}x_1 + 2^{-5}x_3[-1] + 2^{-9}x_2[-1] + 2^{-12}x_1[-1] + 2^{-4}x_5[-2] + 2^{-8}x_3[-2] + 2^{-11}x_2[-2] + 2^{-8}x_4[-3] \quad (4.3)$$

9 LOs are required for (4.3) and 4 LOs to realize x_2, x_3, x_4 and x_5 . Thus a total of 13 LOs are needed to realize the symmetric first half coefficient set (h_0, h_1, h_2 and h_3) of the 8-tap filter in this example using combined HCSE and VCSE. Though the use of VCSE enables elimination of redundant VCSs in the symmetric first half coefficient set, it would require additional LOs to realize the symmetric second half coefficient set due to the delay differences between the VCSs. This can be illustrated as follows. For the symmetric part of (4.3), using the same VCSs, the expression for output of coefficients h_4, h_5, h_6 and h_7 is given by

$$y_3 = 2^{-4}x_5[-4] + 2^{-8}x_4[-4] + 2^{-8}x_3[-5] + 2^{-11}x_2[-5] + 2^{-1}x_5[-6] + 2^{-5}x_3[-6] + 2^{-9}x_2[-6] + 2^{-12}x_1[-6] + 2^{-7}x_2[-7] + 2^{-10}x_1[-7] \quad (4.4)$$

The expression y_3 cannot be obtained directly from y_2 by a single delay operation due to the differences between the delay amounts of certain terms in these two expressions. To obtain y_3 from y_2 , following operations must be performed where $\rightarrow^{[-k]}$ represents delay operation by k units and $y_2 = y_{2,1} + y_{2,2} + y_{2,3} + y_{2,4} + y_{2,5} + y_{2,6}$:

$$y_{2,1} = 2^{-1}x_5 \rightarrow^{[-6]} 2^{-1}x_5[-6] \quad (4.5)$$

$$y_{2,2} = 2^{-7}x_2 + 2^{-10}x_1 \rightarrow^{[-7]} 2^{-7}x_2[-7] + 2^{-10}x_1[-7] \quad (4.6)$$

$$y_{2,3} = 2^{-5}x_3[-1] + 2^{-9}x_2[-1] + 2^{-12}x_1[-1] \rightarrow^{[-5]} 2^{-5}x_3[-6] + 2^{-9}x_2[-6] + 2^{-12}x_1[-6] \quad (4.7)$$

$$y_{2,4} = 2^{-4}x_5[-2] \rightarrow^{[-2]} 2^{-4}x_5[-4] \quad (4.8)$$

$$y_{2,5} = 2^{-8}x_3[-2] + 2^{-11}x_2[-2] \rightarrow^{[-3]} 2^{-8}x_3[-5] + 2^{-11}x_2[-5] \quad (4.9)$$

$$y_{2,6} = 2^{-8}x_4[-3] \rightarrow^{[-1]} 2^{-8}x_4[-4] \quad (4.10)$$

From the above expressions, it is clear that additional LOs are needed to realize

the symmetric second half of coefficients from the first half coefficients – this is the drawback of employing VCSs in [47]. Thus additional LOs would be required to obtain y_3 , i.e., sum of expressions (4.5) to (4.10). Therefore, a total of 19 LOs (12 LOs and 7 inter-tap adders) and an LD of 4 adder-steps would be required to implement the complete 8-tap filter in Table 4.1 using BSE as can be noted from Fig. 4.2. The LD can be reduced to 3 by following binary tree representation [45], but the trade-off will be increased number of LOs due to the inefficiency in exploiting the symmetry of FIR filter.

Figure 4.2 Filter realization using BSE.

The proposed BCSE algorithm groups only bits that occur within each coefficient called as horizontal common subexpression (HCSs) and hence the symmetry property of FIR filter coefficients can be completely exploited. Using the proposed BCSE-4, the output y_4 corresponding to coefficients h_0, h_1, h_2 and h_3 can be expressed as

$$y_4 = 2^{-1}x_1 + 2^{-7}x_7 + 2^{-1}x_1[-1] + 2^{-5}x_3[-1] + 2^{-9}x_7[-1] + 2^{-4}x_1[-2] + 2^{-8}x_3[-2] + 2^{-11}x_2[-2] + 2^{-4}x_1[-3] + 2^{-8}x_4[-3] \quad (4.11)$$

where $x_7 = [1 \ 1 \ 0 \ 1]$ and x_2 to x_4 are HCSs defined earlier. The schematic diagram is shown in Fig 4.3. Thus only 17 LOs are required to realize the example filter in Table 4.1 using BCSE-4 (9 LOs for (4.11), 4 LOs for the HCSs x_2 , x_3 , x_4 , x_7 and 4 inter-tap LOs for the symmetric second half coefficients). The expression for the second symmetric half can be obtained using a single delay operation on (4.11) as VCSs are excluded in BCSE-4 algorithm. The LD required for implementing the filter is 3 adder-steps.

Figure 4.3 Filter realization using proposed BCSE-4.

Using the equation for LD, the LDs of 4-bit CSs and 3-bit CSs are the same (2 adder-steps). Note that the LD of the largest 3-bit CS, $[1 \ 1 \ 1] = x_1 + 2^{-1}x_1 + 2^{-2}x_1$ is 2 adder-steps, and that of the largest 4-bit CS, $[1 \ 1 \ 1 \ 1] = x_1 + 2^{-1}x_1 + 2^{-2}x_1 + 2^{-3}x_1$ is also 2

adder-steps. These largest 3-bit and 4-bit CSs represent the worst-case CSs among all the possible 3-bit and 4-bit CSs as they have the longest LDs in respective groups. Clearly, the LDs of the worst-case 3-bit CS and 4-bit CS are the same. Therefore, the proposed 4-bit CSs-based BCSE-4 method will not increase the LD of the multiplier compared to the 3-bit CSs-based BSE [47]. The reason for the better reduction of LOs achieved by BCSE-4 compared to BSE [47] is that the former method has a better coverage of nonzero bits than the latter for the same number of CSs. The consequence of using higher-order CSs (CSs with more than 4 ones in CSs) was also investigated. Though higher-order CSs would provide an enhanced coverage of nonzero bits, the number of LOs required to realize all the possible subexpressions would also increase. The frequency of higher-order BHCSs, i.e., 5-bit and 6-bit BHCSs were analyzed, which are realized basically as supersets of 4-bit BHCSs. The example filters had wordlengths ranging from 8-bit to 24-bit, different passband and stopband specifications ($\{0.021\pi, 0.07\pi\}$, $\{0.10\pi, 0.12\pi\}$, $\{0.15\pi, 0.25\pi\}$, $\{0.37\pi, 0.5\pi\}$, $\{0.6173\pi, 0.6276\pi\}$, $\{0.6836\pi, 0.6973\pi\}$) respectively for different filter lengths ranging from 10-taps to 1180-taps. The reduction of LOs is not found significant for 5-bit or 6-bit CSs (5 or 6 ones in CSs) due to the following facts. First, the probability of repeated occurrence of such higher-order CSs is less in practical filter coefficients. Second, additional LOs are needed to realize higher-order CSs which would increase the hardware overhead. Furthermore, the use of higher-order CSs with number of ones greater than 4, will increase the LD (i.e. LD of $[1\ 1\ 1\ 1\ 1]=3$ which is 1 adder-step more than that of 4-bit CSs). Due to the above disadvantages, higher-order CSs with more than four ones is omitted in the proposed BCSE-4 method.

The advantages of BSCE-4 over BSE can be summarized as follows.

1. BCSE-4 requires only fewer number of LOs as it can group more number of bits (better coverage of nonzero bits) for an identical or sometimes even fewer numbers of CSs compared to BSE.
2. It uses only horizontal CSs and hence symmetry of FIR filter coefficients can be completely exploited.
3. It will have identical LDs with respect to BSE as the maximum LD for 3-bit and 4-bit CSs remain the same.

The proposed BCSE-4 method consists of elimination of horizontal subexpression and super subexpression as explained below. The binary representation of filter coefficients for the specified wordlength is obtained initially.

4.1.1 Binary Horizontal Subexpression Elimination (BHSE)

The BHSE deals with the elimination of redundant Binary HCSs (BHCSs) that occur within a coefficient.

- a. A 4-bit binary representation can form 7 distinct BHCSs, which are [1 1], [1 0 1], [1 1 1], [1 1 1 1], [1 1 0 1], [1 0 1 1] and [1 0 0 1]. Note that the other possible 4-bit patterns [0 0 1 1], [1 1 0 0] and [0 1 1 0] can be implemented using the CS [1 1], and bit patterns [0 1 0 1] and [1 0 1 0] can be implemented using the CS [1 0 1] with simple shift operations (without using extra adders). Also, the patterns [0 1 1 1] and [1 1 1 0] can be obtained using the CS [1 1 1] with shift operations. The implementations [0 0 0 1], [0 0 1 0], [0 1 0 0] and [1 0 0 0] do not require LOs as these patterns have only one nonzero bit. Thus the realization of the 7 distinct BHCSs listed above requires 7 LOs when 4-bit BHCSs are employed. Along with the above subexpressions, higher-order CSs [1 0 0 0 1], [1 0 0 0 0 1], [1 0 0 0 0 0 1], [1 0 0 0 0 0 0 1] etc are also used for which the LD is the minimum. These minimum LD higher-order CSs occur repeatedly in binary representation of filter coefficients and hence they can eliminate more number of redundant BHCSs. The CSs generated for each CSE method needs to be stored in registers (corresponding to the outputs of distinct CSs) and they are reused (shared) in the filter realization. The number of registers used will be proportional to the number of CSs realized. The slight overhead in the number of LOs for the BHCSs in proposed 4-bit BCSE when compared to 3-bit BSE [47] is negligible taking into consideration of the potential of the former method to achieve a better coverage of nonzero bits using less number of BHCSs than the BSE [47] in the case of practical higher-order FIR filters.

- b. While eliminating BHCSs from the binary coded coefficients, the priority has been set to eliminate BHCSs in the following order of decrementing LD: i.e [1 1 1 1], [1 0 1 1], [1 1 1], [1 1], [1 0 1], [1 0 0 1], [1 0 0 0 1], [1 0 0 0 0 1] etc. This order is selected with the objective of eliminating the BCSs which can give increased reduction of LOs. In this method, some extra bits can be grouped together within the subexpression, as the BHCS is 4-bit wide, and thus save LOs compared to BSE [47]. For instance, consider the grouping of the CSs using 3-bit BHCSs as in Fig. 4.4(a). This requires 2 LOs (excluding LOs for BHCSs). Fig. 4.4(b) illustrates the grouping using proposed BCSE-4. In this case only 1 LO (excluding LOs for BHCSs) is needed to realize h_l which is 1 LO less compared to 3-bit BHCSs. The LD required for both BSE [47] and BCSE-4 is equal to 3.

$$0.000\boxed{101}1000\boxed{11}0$$

Figure 4.4 (a) Coefficient h_l grouped using 3-bit BHCSs.

$$0.000\boxed{1011}000\boxed{11}0$$

Figure 4.4 (b) Coefficient h_l grouped using 4-bit BHCSs.

4.1.2 Binary Horizontal Super Subexpression Elimination (BHSSE)

A greater reduction of LOs can be achieved by forming HSSE from HCSs by utilizing the redundant identical shift between HCSs. Multiple BHCSs with identical shifts between them are combined to form Binary Horizontal Super Subexpressions (BHSSs) in BHSSE. The HSSE method for CSD coefficients was initially proposed in [45] and is also used in [47] as Common Binary CSs (CBCSs) for greater reduction of LOs.

The flow diagram for BHSSE is given as

/* $h(m)$ = m -th filter coefficient*/

/* N is the filter length and we need to implement only $\left\lceil \frac{N}{2} \right\rceil$ coefficients as FIR filter

is symmetric*/

Initialize $m=0, n=0$.

1. Identify the BHSSs in $h(m)$;
2. Set $n=m+1$; check if $n \leq \left\lceil \frac{N}{2} \right\rceil$, then go to Step 3, else go to Step 5.
3. Eliminate all the BHSSs in $h(m)$ from $h(n)$; In this repeated BHSS are replaced while keeping the original;
4. Increment n and check whether $n \leq \left\lceil \frac{N}{2} \right\rceil$; If so, go to Step 3, else go to Step 5.
5. Increment m , check whether $m \leq \left\lceil \frac{N}{2} \right\rceil$; If so, go to Step 1, else go to Step 6;
6. Terminate the program.

The BHSSE can be explained using the example of the first symmetric half of the coefficients of an 8-tap filter shown in Table 4.2. The numerals in the first row represent the number of bit-wise right shifts. Note that, 2 represents [1 1], 3 represents [1 1 1], 4 represents [1 0 1] and 6 represents [1 0 0 1]. The BHCSs numbered 4 and 3 with a shift difference of 6 are repeated for h_0 and h_1 and [4 0 0 x x 0 3] forms a BHSS, where x is a don't care. It must be ensured that the subexpressions forming super subexpressions are not modified with the introduction of a subexpression at the position of don't care. The BHSS, [4 0 0 x x 0 3] is replaced by 9 as indicated in Table 4.3. The BHSSs need to be implemented only once and redundant computations can be eliminated.

Table 4.2
Representation of filter coefficients for BHSSE.

	1	2	3	4	5	6	7	8	9	10	11	12
h_0	4	0	0	x	x	0	3	0	0	0	0	0
h_1	0	4	0	0	x	x	0	3	0	0	0	1
h_2	0	0	0	0	2	0	0	0	4	0	0	0
h_3	0	3	0	0	0	1	0	0	2	0	0	0

Table 4.3
Representation of filter coefficients after BHSSE.

	1	2	3	4	5	6	7	8	9	10	11	12
h_0	9	0	0	0	0	0	0	0	0	0	0	0
h_1	0	9	0	0	0	0	0	0	0	0	0	1
h_2	0	0	0	0	2	0	0	0	4	0	0	0
h_3	0	3	0	0	0	1	0	0	2	0	0	0

Hence for the proposed BCSE-4 method both BHSE and BHSSE is done. The BCSE-4 procedure is summarized as follows.

- (1) Obtain the filter coefficients in binary-coded format.
- (2) Eliminate the repetition of BHCSs [1 1], [1 0 1], [1 1 1], [1 0 0 1], [1 1 0 1], [1 0 1 1] and [1 1 1 1] and higher-order CSs with LD=1, with importance given to minimize LD (i.e BHCS [1 1 1 1] is eliminated only when the LD of that particular coefficient remains minimum).
- (3) Eliminate the common BHSS as described in the BHSSE procedure.

4.2 Extension of BCSE-4 to High-Level Synthesis

The adder reductions achieved using BCSE-4 can be better justified from the perspective of high-level synthesis (HLS). In HLS, the primary goal of transformations has been to optimize the dedicated architectures to obtain a good trade-off between area, power, and speed of operation of the circuit [61]. CSE has been used as a powerful HLS transformation to reduce power consumption and area [36], [38]. These approaches focused on reducing the number of operations (additions) and, therefore, area and power of designs. The high-level synthesis tasks of these methods were based on elimination of two nonzero-bit common subexpressions as shown in Fig. 4.5(a). The operands a , b , c , and d in Fig. 4.5(a) represent the input signal of the filter and its shifted versions. The sums e and f are the CSs that are shared in minimizing adders, and S_1 , S_2 , S_3 , and S_4 represent the number of right shifts. Note that four adders are required to obtain the final expressions g and h . The binary representation-based CSE technique in [47] which

employs 2-bit and 3-bit CSE also follows a HLS transformation similar to the scheme shown in Fig. 4.5(a). Fig. 4.5(b) illustrates the BCSE-4 method where the super-subexpression i , (formed from CSs e and f) is shared for further reduction of adders to obtain g and h using appropriate shifts (s_3 and s_4 respectively). Thus 3 adders are needed using BCSE-4, which is 1 adder less than the conventional CSE shown in Fig. 4.5(a). For higher-order filters, the number of super-subexpressions will be more and their sharing would offer reduction of adders. In BCSE-4, the number of operations (LOs) is reduced through the reduction of adders by employing the 4-bit CSs, which is a super set of the horizontal CSs employed in BSE [47]. Note that the BCSE-4-based HLS transformation is similar to the super-subexpression elimination in [45], but the difference is that the proposed method uses binary-coded coefficients whereas [45] uses CSD coefficients. It has been shown in [47] that binary representation will offer better adder reductions than CSD. Therefore, BCSE-4 is a better method than [45]. The LD is kept low in the proposed method by following the binary tree structure [45] architecture. Thus, by employing the proposed transformation (4-bit subexpressions, which are super-subexpressions formed from 2-bit and 3-bit CSs), BCSE-4 improves the efficiency of CSE in high-level synthesis and offers a power efficient solution by reducing the number of operations (additions).

Fig. 4.5 Subexpression sharing as a high-level synthesis transformation.
(a) Conventional 2-bit CSE. (b) Proposed 4-bit CSE (BCSE-4).

4.3 Design Examples

Several FIR filter design examples are presented in this section to show the complexity comparison of various works in literature with the proposed work. The filter coefficients are generated using the Parks-McClellan algorithm.

Example I: In this example, the complexity reductions in terms of LOs and LDs, of five benchmark FIR filters FIR1 to FIR5, extensively used in literature are compared. FIR1 and FIR2 are the filters presented in design examples presented in [33]. FIR1 is a 25-tap filter with passband frequency of 0.15π , stopband frequency of 0.25π and wordlength of 9-bits. FIR2 is a 59-tap, 14-bit wordlength filter with the passband and stopband frequencies of 0.021π and 0.070π respectively. FIR3 is a 120-tap high pass filter with 17-bit wordlength, stopband and passband frequencies of 0.37π and 0.5π respectively. FIR4 is a 200-tap, 13-bit wordlength filter employed in the filter bank channelizer of Digital Advanced Mobile Phone Systems (D-AMPS) with passband and stopband frequencies of 0.6173π and 0.6276π respectively. FIR5 is the filter employed in the receivers for the Personal Digital Cellular (PDC) standard. It is a 230-tap, 16-bit wordlength filter with passband and stopband frequencies at 0.6836π and 0.6973π respectively. The comparison of LOs and LDs is tabulated in Table 4.4. The proposed BCSE-4 is compared with Pasko [39], NR-SCSE [40], RAG- n [49], Chia [41], CRA [51], H-CUB [52] and BSE [47]. In Table 4.4, N represents the filter length and W , the wordlength. The online version of the H-CUB algorithm [52] supported only for filter length lesser than 200 and hence those results which are not available are indicated as NA in Table 4.4.

From Table 4.4, it can be noted that the proposed BCSE-4 offers an overall average reduction of 13.5% in LOs compared to Pasko [39], 7.36% compared to NR-SCSE [40] and 15.5% compared to CRA [51]. The results after applying BCSE-4 are slightly inferior to H-CUB [52] for FIR1 to FIR3 in terms of LOs, but the LDs of the BCSE-4 are substantially lower than that of the H-CUB algorithm. As the number of taps increases (120-taps and more), the proposed BCSE-4 outperforms H-CUB [52] in number of LOs also. The proposed algorithm offers the best reduction of LOs and LDs for higher-order and larger wordlength filters as can be seen from the comparison with RAG- n [49] and Chia [41]. Also, RAG- n [49] has an upper limit for coefficient wordlength due to the use of look up tables. An average LO reduction of 6.4% is obtained using the proposed BCSE-4 method when compared to BSE [47], and the LDs of these two methods are identical. The proposed BCSE-4 maintains a good trade-off between LOs and LDs by reducing the LOs and without increasing the LDs, especially for higher-order filters.

Table 4.4
Comparison of LOs and LDs for benchmark FIR filters in example I.

Filter			Pasko[39]		NR-SCSE[40]		RAG-n[49]		Chia[41]		CRA[51]	
	N	W	LO	LD	LO	LD	LO	LD	LO	LD	LO	LD
FIR1	25	9	6	2	6	2	6	2	6	2	6	2
FIR2	59	14	32	2	30	2	26	2	26	2	26	2
FIR3	120	17	58	3	58	3	51	4	51	3	55	3
FIR4	200	13	68	4	68	3	64	5	70	3	70	3
FIR5	230	16	139	4	137	3	127	5	127	4	138	3

Filter			H-CUB[52]		BSE[47]		Proposed BCSE-4	
	N	W	LO	LD	LO	LD	LO	LD
FIR1	25	9	5	2	6	2	6	2
FIR2	59	14	24	6	29	3	29	3
FIR3	120	17	48	8	50	4	53	4
FIR4	200	13	63	6	60	3	58	3
FIR5	230	16	NA	NA	120	4	102	4

Example II: In this example, the FIR filters S1 and S2, used in the design examples of [32] are considered. The passband and stopband frequencies of S1 and S2 are 0.10π and 0.12π respectively. S1 is 100-tap, 16-bit wordlength filter whereas S2 is 120-tap, 18-bit wordlength filter. Table 4.5 shows the comparison of LOs and LDs for filters realized using BCSE-4 and the methods in [36] and [32]. The results of [32] and [36] are taken from [32]. The comparison of LOs and LDs with CRA [51] and BSE [47] are shown in Table 4.5. The average percentage reduction of LOs using BCSE-4 method is 4.6% over MSD based [32], 18.05% over CSD-based [32], 27.8% over [51] and 7.02% over [47]. It can be noted that the LDs obtained using BCSE-4 are identical to LDs of [47]. BCSE-4 shows an LD reduction of the filter about 1 to 2 adder-steps when compared to [32].

Table 4.5
Comparison of LOs and LDs of filters in example II.

Filter	Potkjak[36]		MSD based[32]		CSD based[32]	
	LO	LD	LO	LD	LO	LD
S1	95	5	70	4	83	4
S2	136	6	91	4	104	5

Filter	CRA[51]		BSE[47]		Proposed BCSE-4	
	LO	LD	LO	LD	LO	LD
S1	74	3	76	4	65	4
S2	101	3	91	4	89	4

Example III: In this example, the hardware complexities of the FIR filter with the passband and stopband frequencies of 0.2π and 0.22π respectively are compared. The LOs and LDs for different filter lengths of 20, 50, 80, 120, 200 and 400 and for different wordlengths of 12, 16, 20 and 24 bits are compared in this example. Tables 4.6 and 4.7 show the comparison of the LOs and LDs respectively needed to implement the filter using BCSE-4 method and the methods in [40], [51], [52] and [47].

Table 4.6 shows that H-CUB [52] and CRA [51] offer better reduction of LOs compared to the BCSE-4 for lower-order filters. From Table 4.7, it can be noted that the LDs of filters realized using H-CUB are substantially longer than that of BCSE-4. Also, the results of H-CUB are available only up to 200 taps. If filters with taps more than 200 were realized using H-CUB, clearly it would result in multipliers with very longer LDs (more than 9 adder-steps), which is not suitable for high-speed filtering applications. For lower-order filters, binary representation-based CSE techniques (BSE [47] and proposed BCSE-4) fail to offer good reduction of LOs compared to CSD-based CSE methods because of following reasons [47]. First, the number of BHSSs will be less for lower-order filters (In general, the number of BHSSs is proportional to filter order). Second,

after applying BCSE technique, more number of unpaired bits occur for lower-order filters compared to higher-order filters. When compared with BSE [47], the BCSE-4 is able to achieve better reduction of LOs for higher-order filters as more number of bits can be grouped together to form CSs, which in turn result in fewer number of unpaired bits.

Table 4.6
Number of LOs for the filters in example III.

Filter Length (N)	NR-SCSE[40]				CRA[51]				H-CUB [52]			
	12 bit	16 bit	20 bit	24 bit	12 bit	16 bit	20 bit	24 bit	12 bit	16 bit	20 bit	24 bit
20	18	24	37	49	19	23	34	41	13	17	24	29
50	25	48	67	87	27	45	66	86	20	37	45	65
80	50	72	91	126	46	67	90	121	33	59	80	119
120	50	79	134	179	46	78	123	165	34	72	110	159
200	50	126	179	250	46	121	176	241	38	100	164	225
400	55	186	305	450	51	174	294	421	NA	NA	NA	NA

Filter Length (N)	BSE[47]				Proposed BCSE-4			
	12 bit	16 bit	20 bit	24 bit	12 bit	16 bit	20 bit	24 bit
20	17	24	31	40	15	24	30	40
50	26	48	58	72	25	42	61	72
80	34	61	78	112	33	55	77	100
120	31	69	107	157	28	66	101	134
200	34	91	159	213	33	79	159	198
400	40	126	226	348	40	126	226	342

The LDs of the BCSE-4 are identical to that of BSE [47]. When compared with NR-SCSE [40] and CRA [51], the LDs of BCSE-4 is slightly longer for 12-bit coefficients and identical for 16, 20 and 24 bit coefficients. The average reduction of LOs achieved using BCSE-4 over NR-SCSE [40] is 20.2% and over CRA [51] is 17.5% respectively for higher-order filters. Thus the proposed BCSE-4 method offers the best trade-off between the LOs and LDs especially for higher-order filters. Fig. 4.6 shows the reduction of LOs achieved using BCSE-4 and the CSE methods [40], [51], [52] and [47], over the direct method for filters of lengths 20, 50, 80, 120, 200 and 400. The coefficient wordlength is 24-bits. The BCSE-4 offers an average LO reduction of 11% over the NR-

SCSE [40], 8% over CRA [51], 6% over BSE [47] and 4.5% over H-CUB [52]. Though the BCSE-4 offers only a slight reduction of LOs over H-CUB, it produces filters with substantially shorter LDs compared to latter. Moreover, the applicability of H-CUB is not known for filters with more than 200 taps. Clearly, the LDs of such higher-order filters realized using H-CUB will be too long (more than 9 adder-steps), which would affect the speed of the filter adversely. Note that the proposed BCSE-4 method offers better reduction of LOs for filters with 80-taps and more. Generally, the BCSE-4 method becomes more effective for higher-order filters.

Table 4.7
Number of LDs for the filters in example III.

Filter Length (N)	NR-SCSE[40]				CRA[51]				H-CUB [52]			
	12 bit	16 bit	20 bit	24 bit	12 bit	16 bit	20 bit	24 bit	12 bit	16 bit	20 bit	24 bit
20	2	3	4	4	2	3	4	4	6	6	7	8
50	2	3	4	4	2	3	4	4	6	6	7	8
80	2	3	4	4	2	3	4	4	6	6	7	8
120	2	3	4	4	2	3	4	4	7	7	8	9
200	2	3	3	4	2	3	3	4	8	8	8	9
400	2	3	4	4	2	3	4	4	N	NA	NA	NA

Filter Length (N)	BSE[47]				Proposed BCSE-4			
	12 bit	16 bit	20 bit	24 bit	12 bit	16 bit	20 bit	24 bit
20	3	3	4	4	3	3	4	4
50	3	3	4	4	3	3	4	4
80	3	3	4	4	3	3	4	4
120	3	3	4	4	3	3	4	4
200	3	3	4	4	3	3	4	4
400	3	3	4	4	3	4	4	4

Example IV: The proposed BCSE-4 method was used for realizing FIR filters employed in the filter bank channelizer of D-AMPS [62]. The channel filters extract 30 kHz D-AMPS channels from the input signal after down sampling by a factor of 350. The passband and stopband edges are 30 and 30.5 kHz respectively. The peak passband ripple

is chosen as 0.1 dB. The filter stopband specifications are chosen as in the D-AMPS standard [62]. The FIR filter length N is determined using the following equation [22]:

Figure 4.6 Percentage reduction of LOs for 24-bit wordlength filter used in example III using NR-SCSE [40], H-CUB [52], BSE [47], CRA [51] and proposed BCSE-4 over direct method.

$$N = \frac{-10 \log_{10} \partial_1 \partial_2 - 13}{14.6 \Delta f} + 1 \quad (4.12)$$

where ∂_1 and ∂_2 are the peak passband and stopband ripples respectively, and Δf is the normalized width of the transition band. The comparison of LOs and LDs needed to implement the filter using the proposed BCSE-4 method with NR-SCSE [40], Super-Subexpression (SS) method [45], BSE [47] and CRA [51] are shown in Tables 4.8 and 4.9 respectively.

Comparison was done for wordlengths of 12, 16, 20 and 24 bits for filter lengths of 460, 610, 940 and 1180 corresponding to peak stop band ripples (PSR) of -48 dB, -65 dB, -85 dB and -96 dB respectively obtained using (4.12). The comparison with H-CUB [52] is not shown because the maximum length of filters that can be realized using H-CUB is 200, as per the H-CUB filter synthesis algorithm available [52].

Table 4.8
Comparison of LOs for D-AMPS filters in example IV.

Filter Length (N)	NR-SCSE[40]				CRA[51]				SS [45]			
	12 bit	16 bit	20 bit	24 bit	12 bit	16 bit	20 bit	24 bit	12 bit	16 bit	20 bit	24 bit
460	55	250	370	500	55	246	361	497	52	203	350	500
610	59	287	480	620	53	268	457	613	50	253	467	620
940	55	325	640	870	55	322	623	859	50	300	635	855
1180	67	374	723	1040	65	358	700	1032	55	373	706	987

Filter Length (N)	BSE[47]				Proposed BCSE-4			
	12 bit	16 bit	20 bit	24 bit	12 bit	16 bit	20 bit	24 bit
460	41	149	274	408	34	159	252	390
610	42	164	316	473	33	164	306	471
940	42	181	384	636	36	177	370	587
1180	46	203	425	707	37	181	419	645

Table 4.9
Comparison of LDs for D-AMPS filters in example IV.

Filter Length (N)	NR-SCSE[40]				CRA[51]				SS [45]			
	12 bit	16 bit	20 bit	24 bit	12 bit	16 bit	20 bit	24 bit	12 bit	16 bit	20 bit	24 bit
460	2	3	4	5	2	3	4	5	3	3	4	5
610	2	4	4	5	2	4	4	5	3	3	4	5
940	2	3	4	5	2	3	4	5	3	3	4	5
1180	2	3	5	5	2	3	5	5	3	4	4	5

Filter Length (N)	BSE[47]				Proposed BCSE-4			
	12 bit	16 bit	20 bit	24 bit	12 bit	16 bit	20 bit	24 bit
460	3	3	4	4	3	3	4	4
610	3	3	3	4	3	3	4	4
940	3	3	4	4	3	3	4	4
1180	3	4	4	4	3	4	4	4

From Table 4.8, the percentage reduction of LOs using BCSE-4 for filters with coefficient wordlength of 24-bit is 29.1% over [40], 27.9% over [45], 6.5 % over [47] and 28.4% over [51]. Table 4.9 shows that the LDs for the proposed method are same as that of SS [45], BSE [47] (except for 610-tap, 20-bit wordlength, where LD is one adder-step more). The LDs of filters with 12-bit coefficient wordlength is least for NR-SCSE [40] and CRA [51]. For the 16-bit coefficients, the LDs of NR-SCSE and BCSE-4 are identical whereas for the 20-bit and 24-bit coefficients, the BCSE-4 has shorter LDs compared to NR-SCSE.

Figure 4.7 Percentage reduction of LOs for 940-tap D-AMPS filter using NR-SCSE [40], SS [45], BSE [47], CRA [51] and proposed BCSE-4 over direct method.

Fig. 4.7 shows the comparison of reductions of LOs for a 940-tap D-AMPS filter, for various wordlengths using BCSE-4, NR-SCSE [40], SS [45], BSE [47] and CRA [51] over direct method of implementation. The reductions of LOs achieved using BCSE-4 are 13.43% over [40], 11.9% over [45], 1.5% over [47] and 13% over [51]. The proposed method has the best percentage reduction of LOs compared to [40], [45], [51] and [47]. Fig. 4.8 shows the reduction in LOs for a 16-bit D-AMPS filter for various filter lengths. From Fig. 4.8, the average reductions of LOs achieved using BCSE-4 are 18.075% over [40], 14.1 over [45], 0.5% over [47] and 16.7% over [51]. It can be noted that the

proposed BCSE-4 results in 8.1% improved LO reduction over [47] for higher-order ($N > 900$) filters.

Figure 4.8. Percentage reduction of LOs for 16-bit wordlength D-AMPS filter using NR-SCSE [40], SS [45], BSE [47], CRA [51] and proposed BCSE-4 over direct method.

4.4 Synthesis and Implementation Results

The cost metrics (power, area and delay) for filters realized using proposed BCSE-4 and other CSE methods were analyzed and compared using state of the art CAD tools. Filters were synthesized using TSMC 0.18 μ m technology for 8-bit quantized input signal. The area and delay are measured directly using the synopsis design compiler. The power is calculated by inputting a set of random numbers (8-bit in length) through a test bench program and the average value is reported. The examples taken are T1, T2, T3 and T4, where T1 is a 30-tap FIR filter with passband frequency of 0.15π , stopband frequency of 0.25π and wordlength of 16-bits, T2 is a 60-tap FIR filter with passband frequency of 0.021π , stopband frequency of 0.07π and wordlength of 14-bits, T3 is a 90-tap FIR filter with passband frequency of 0.15π , stopband frequency of 0.25π and wordlength of 14-bits and T4 is a 140-tap FIR filter with passband frequency of 0.6173π , stopband frequency of 0.6276π and wordlength of 14-bits. The comparison of synthesis results for various CSE algorithms with the BCSE-4 is given in Table 4.10. For T1, the binary representation-

based CSE methods (BSE [47] and BCSE-4) gave inferior results as the number of taps is less. For shorter filters, the factor N_{up} in Section 4.1, is almost same for CSD and Binary representations and hence CSD-CSE methods will result in comparable or better adder reduction [47]. As expected, NR-SCSE produces minimum delay filters for lower-order filters which can be seen clearly from all the examples in Table 4.10.

Table 4.10
Comparison of synthesis results for filters T1, T2, T3 and T4 using 0.18 μ m TSMC technology.

Filter	CSE algorithms	Area in μm^2	Power in mW	Delay in ns
T1	NR-SCSE[40]	76962.5	3.9814	6.205
	BSE[47]	100416	4.4052	6.87
	PROPOSED BCSE-4	87726.39	4.0902	6.87
T2	NR-SCSE[40]	133645	7.2037	6.02
	BSE[47]	121787.41	7.0037	6.22
	PROPOSED BCSE-4	120856.88	6.8746	6.22
T3	NR-SCSE[40]	178474.56	9.9475	6.24
	BSE[47]	176076.81	9.7103	6.24
	PROPOSED BCSE-4	174007.65	9.5304	6.24
T4	NR-SCSE[40]	275785.86	12.21501	6.815
	BSE[47]	248718.2	11.9603	6.815
	PROPOSED BCSE-4	229653.1	11.2104	6.815

For T2, T3 and T4, the BCSE-4 results in reduced area and power compared to CSD-based NR-SCSE [40]. The delay and power increases for BSE [47] compared to proposed BCSE-4 due to the fact that the use of VCSs in BSE results in overhead of LOs to realize the symmetric part of coefficients. The BCSE-4 achieves power reductions of 5.2% and 6.1% over the methods in [47] and [40] respectively. As the number of taps and wordlengths increases, binary representation-based CSE methods produced filters with fewer LOs compared to CSD based CSE methods. From the synthesis results, it is evident that the BCSE-4 algorithm produces FIR filters with low power consumption.

4.5 Summary

In this chapter, a new 4-bit Binary based Common Subexpression Elimination algorithm (BCSE-4) to realize FIR filters with fewer number of LOs and low-power consumption without increasing the delay is proposed. The advantage of the proposed BCSE-4 method over previously proposed 3-bit binary subexpression elimination (BSE) algorithm [47] is that it offers a better coverage of nonzero bits using less number of subexpressions, resulting in reduced number of LOs. In addition, it was found that there is almost no increase in logic depth for the proposed BCSE-4 method. The reduction of adders achieved using the proposed BCSE-4 method is slightly inferior for short filters. However, the BCSE-4 method offers better reduction of adders without any increase in the critical path length for higher-order filters. Therefore, the proposed BCSE-4 method is best suited for implementing higher-order FIR filters in software defined radio (SDR) channelizers. The BCSE-4 method gives average power reductions of 6.1% and 5.2 % over the methods in [40] and [47] respectively, when synthesized with TSMC-0.18 μ m technology. Design examples show that, for a wordlength of 24-bit, BCSE-4 offers an overall LO reduction of 29.1% compared to [40], 27.9% compared to [45], 6.5 % compared to [47] and 28.4% compared to [51]. However the proposed BCSE-4 method and all the CSE techniques [37]-[47] in literature have been applied only to fixed-coefficient FIR filters. In the next chapter, the reconfigurable FIR filters for realizing channel filters for SDR are proposed.

Chapter 5

Reconfigurable Low Complexity Filters and Filter Banks based on Multi-band Filtering and Masking Technique

In this chapter, reconfigurable low complexity filters and filter banks based on a multi-band filtering and masking technique is proposed. The channel filters based on conventional filter design techniques will have large number of filter taps, N , as per equation (4.12). Although complexity reduction techniques can be applied to fixed coefficient filters designed using conventional techniques such as fixed coefficient filter optimizations would affect the reconfigurability of the filter adversely. In order to extract non-uniform channels, the conventional per-channel (PC) method uses separate channel filters corresponding to the frequency response specifications of the channels of different wireless communication standards. Reconfigurability is achieved in PC approach by switching among distinct filter according to the current standard of operation. This is not an efficient approach due to its increased hardware complexity and poor resource utilization as mentioned in Section 2.5.1 of Chapter 2. Hence new techniques are needed to be explored to reduce the complexity and also to incorporate reconfigurability to the SDR channel filter so as to extract non-uniform channels. In this chapter two such methods are proposed.

The first method is based on multi-band filtering and masking technique. The front-end of the proposed multi-band filtering and masking architecture employs a multi-band filter where each frequency band corresponds to the channel of distinct communication standard. Based on the current standard of operation, the desired channel is extracted from the multi-band output using appropriate masking filters at the back-end. For a specific set of wireless standards, as the multi-band filter at the front-end is a common filter, it can be implemented as a fixed-coefficient filter. Therefore its complexity can be reduced using the low complexity filter realization method explained in Chapter 4. The back-end can be reconfigured according to the current standard of

operation using appropriate masking filters. The order of the multi-band filter is the filter order corresponding to the channel that has the most stringent frequency response specification among the frequency response specifications of all the channels for which the multi-band filter is designed. It can be noted that the PC method needs distinct filters for each frequency band, thus making it highly complex than that of multi-band and masking method. The masking filters have large transition bandwidth and hence are of lower order, compared to that of multi-band filter. Thus the total number of filter taps in the proposed multi-band and masking method is substantially less than that in PC method.

The second method is based on multi-stage FRM (MSFRM) technique. The FRM technique was originally proposed for the design of sharp transition-band FIR filters with very low complexity [63]. The basic idea was to compose the over-all sharp transition-band filter using three wide transition-band subfilters. Since the subfilters have wide transition-band specifications, they can be of lower-order. Consequently, the over-all complexity of FRM-based filter will be much less than sharp transition-band FIR filters designed using conventional methods. The proposed architecture is unique in a way that it is able to effectively exploit the redundancy in multi-stage realization by utilizing the common masking filters thus achieving very low complexity. The work in this chapter also focuses on integrating reconfigurability into the MSFRM-based filter. This chapter also proposes a reconfigurable filter bank based on MSFRM which overcomes certain shortcomings of existing filter banks such as simultaneous extraction of non-uniform bandwidth channels, extremely narrowband channels and multi-mode operation. The proposed filter and filter bank have the flexibility of changing frequency responses dynamically with very low hardware overhead.

5.1 Proposed Multi-band Filter and Masking Method

In this section, a low complexity reconfigurable channel filter architecture using multi-band filter and masking methods is proposed. The proposed channel filter uses the same front-end filter (multi-band filter) for all the wireless standards and channel adaptation is accomplished using a low complex reconfigurable masking filter at the back-end. Each of the wireless communication standards has fixed channel bandwidth which is known

beforehand. Using the proposed approach the front-end multi-band filter is designed in such a way that each channel represents the channel bandwidth (bandwidth) of a distinct communication standard. Fig. 5.1 shows the block diagram of the proposed channel filter architecture. The multi-band filter ($H_a(z)$) splits the input wideband signal into i -frequency bands (channels), each channel represent the channel bandwidth (bandwidth) of a distinct communication standard. Each frequency band of the multi-band filter is fixed according to the channel bandwidth and stopband attenuation specifications of the chosen communication standard. The channel-of-interest from the multiple frequency bands of the multi-band filter is extracted by a suitably designed reconfigurable masking filter ($H_{mai}(z)$).

Figure 5.1 Block diagram of proposed multi-band and masking based channel filter architecture.

The proposed approach can be explained with the help of Fig. 5.2. Fig 5.2 shows three channels CH-1, CH-2 and CH-3 corresponding to three distinct standards S_1 , S_2 and S_3 respectively. Fig. 5.2(a) shows the frequency response of the multi-band filter $H_a(z)$. The passband and stopband edges of the multi-band filter are f_{p1} and f_{s1} respectively for the lowpass band CH-1; stopband edges for the bandpass response CH-2 are f_{s2} and f_{s3} , and corresponding passband edges are f_{p2} and f_{p3} ; stopband edges for the bandpass response CH-3 are f_{s4} and f_{s5} , and corresponding passband edges are f_{p4} and f_{p5} . Channel extraction of CH-1, CH-2 and CH-3 are accomplished by using reconfigurable masking filters $H_{ma1}(z)$, $H_{ma2}(z)$ and $H_{ma3}(z)$ respectively, whose frequency responses are indicated using thin lines in Figures 5.2(b)-5.2(d). The responses in bold lines shown in Figures 5.2(b)-5.2(d), $H_{CH-1}(z)$, $H_{CH-2}(z)$ and $H_{CH-3}(z)$, show the outputs of the masking filter, i.e., $H_{CH-1}(z)$ in 5.2(b) is the overall response when the multi-band filter in 5.2(a) is masked with $H_{ma1}(z)$ and 5.2(c) is the overall response when the multi-band filter is masked with $H_{ma2}(z)$ and so on.

Figure 5.2 Frequency response illustrations of multi-band filtering and masking based channel filter.

A common fixed-coefficient multi-band filter can be employed at the front-end, which can be implemented using low complexity filter realization techniques. The output of the multi-band filter is fed to a reconfigurable masking filter. This procedure will help to adapt to the bandwidth of the desired channel, based on the channel bandwidth and the centre frequency of desired channel, by masking the frequencies of other unused channels.

If conventional PC architecture is employed for channel extraction, three separate filters of lengths N_1 , N_2 , and N_3 corresponding to the frequency response specifications of *CH-1*, *CH-2* and *CH-3* would be required. The length of the multi-band filter is chosen so as to satisfy the most stringent transition-band and stopband attenuation specifications among the specifications of all the standards under consideration. For example, if the transition-band width is the smallest and the stopband attenuation is the largest for *CH-3*, and assuming these specifications relax in the sequential order for *CH-2* and *CH-1*, i.e., *CH-1* has the widest transition-band width and lowest stopband attenuation, then the lengths of the 3 filters needed to extract these channels are related by the expression $N_3 >$

$N_2 > N_1$, if PC architecture is employed. In the proposed architecture, N_3 (filter length corresponding to the most stringent communication standard S_3) is fixed as the length of the multi-band filter. This is because the longest filter will offer the best stopband attenuation and the steepest transition-band. Therefore, the same multi-band filter of length N_3 will satisfy the relatively relaxed transition-band width and/or stopband attenuation specifications of the standards S_1 and S_2 . The masking filters $H_{ma1}(z)$, $H_{ma2}(z)$ and $H_{ma3}(z)$ are designed with wider transition-bands as shown in Fig. 5.2 so that only shorter filters are needed to mask the unwanted frequency bands. The masking filters are implemented using look-up tables (LUTs) and they can be reconfigured by just swapping the LUT values. The size of the LUT would be small as the length of the masking filter is substantially less than that of the multi-band filter. Typically, the total masking filter length is less than 5% of the multi-band filter length, as shown in the design examples in Section 5.2.1. In order to adapt to the channels-of-interest, the multi-band filter and the masking filter responses are frequency shifted appropriately, similar to that of PC approach.

5.2 Experimental Results

5.2.1 Design Examples

Example 1: In this section, the design example of a reconfigurable multi-band and masking based channel filter architecture for a dual-standard Time Division Multiple Access (TDMA) IS-54/IS-136 [64]/ Personal Digital Cellular (PDC) specifications is presented. The sampling rate chosen is 34.02 MHz [62] and a bandwidth of 17.01 MHz is considered. The channel bandwidth of PDC standard is 25 kHz and that of TDMA is 30 kHz, and a down sampling factor of 180 is used. The peak stopband ripple (δ_s) is taken as -40 dB for TDMA and -60 dB for PDC and peak passband ripple (δ_p) is taken as 0.1 dB for both the standards. The passband and stopband edge specifications are TDMA: $f_{p1} = 30$ kHz and $f_{s1} = 30.5$ kHz, PDC: $f_{p2} = 25$ kHz and $f_{s2} = 25.5$ kHz. The filter length N is computed using (5.1) [65]:

$$N = \frac{-2 \log_{10}(10\delta_p\delta_s)}{3(f_s - f_p)} - 1 \quad (5.1)$$

The filter lengths for TDMA and PDC are obtained as 785 and 1051 respectively as shown in Table 5.1. The length of the multi-band filter is fixed as 1051 corresponding to the most stringent transition-band width specification. The magnitude responses of the multi-band filter and the masking filters are shown in Fig. 5.3. The masking filters have large transition-band width and consequently, they are shorter in length. The lengths of the masking filters for TDMA and PDC computed using (5.1) are 14 and 26 respectively.

Table 5.1
TDMA/PDC filter specifications.

Standard	Passband edge (f_p)	Stopband edge (f_s)	N
TDMA	$(30 \times 180) / 34020$ $= 0.159$	$(30.5 \times 180) / 34020$ $= 0.1615$	785
PDC	$(25 \times 180) / 34020$ $= 0.1325$	$(25.5 \times 180) / 34020$ $= 0.135$	1051

Figure 5.3 Magnitude response of multi-band filter and the corresponding masking filters (TDMA/PDC).

The overall filter length needed for the dual-standard operation in the proposed multi-band and masking architecture (N_{pa}) is $1051 + 14 + 26 = 1091$, whereas conventional PC architecture would require 785-tap filter for TDMA and 1051-tap filter for PDC: - Thus the total filter length (N_{pc}) is $785 + 1051 = 1836$. It can be noted that only one of the two separate filters is needed at a time in PC architecture and therefore the

substantial reduction of overall filter length in the proposed architecture may not translate to proportional savings in power consumption. Nevertheless, the multi-band and masking architecture consumes only considerably less area, which is proportional to the reduction of overall filter length, when compared to PC architecture.

Example II: Another dual-standard operation based on CDMA/WCDMA is given for illustration. The sampling rate chosen is 20 MHz. The channel bandwidth of CDMA and WCDMA are 1250 kHz and 5 MHz respectively [64]. The passband and stopband edges are 1250 and 1300 kHz respectively for CDMA and 5000 and 5050 kHz for WCDMA. The values of δ_p and δ_s are chosen as 0.1 dB and -48 dB respectively for both the standards. The filter length computed using (5.1) is identical (858 taps) for CDMA and WCDMA standards because both the standards have identical transition-band width and adjacent channel attenuation (stopband) requirements. Therefore, the length of the multi-band filter in the proposed architecture is fixed as 858. On the other hand, conventional PC architecture would require twice this filter length as two separate filters are used. The magnitude responses of the multi-band filter and the masking filters are shown in Fig. 5.4. The reconfigurable masking filters need only filter lengths of 18 each for masking CDMA and WCDMA outputs. From the filter lengths, it is evident that the complexity of the architecture is approximately half of that of PC architecture in this example.

Figure 5.4 Magnitude response of multi-band filter and the corresponding masking filters (CDMA/WCDMA).

5.2.2 Functionality Test

The channelization functionality of the proposed multi-band filter and masking architecture for the CDMA/WCDMA design example is verified using the following experiment. The specifications are same as that of Example II. A modulated audio waveform signal with a bandwidth of 10 MHz is provided as the input to the architecture. Fig. 5.5(a) shows the frequency spectrum of the input signal. Fig. 5.5(b) shows the spectrum at the output of multi-band filter.

Figure 5.5(a) Input spectrum.

Figure 5.5(b) Output spectrum of Multi-band filter.

After applying masking, the CDMA and WCDMA output spectra are extracted as shown in Fig. 5.5(c) and Fig. 5.5(d) respectively. The spectrum of CDMA satisfies the design specifications ($f_p=1250$ kHz and $f_s=1300$ kHz $\delta_p = 0.1$ dB and $\delta_s = -48$ dB) and WCDMA satisfies the design specifications ($f_s = 3950$ kHz, $f_p = 4000$ kHz and $f_p = 9000$ kHz, $f_s = 9050$ kHz and $\delta_p = 0.1$ dB and $\delta_s = -48$ dB) as shown in Figures 5.5(c) and 5.5(d). It can be noted that using this method, channels corresponding to various communication standards can be effectively extracted.

Figure 5.5(c) CDMA output spectrum using multi-band filter and masking based architecture.

Figure 5.5(d) WCDMA output spectrum using multi-band filter and masking based architecture.

5.2.3 Implementation Results

The dual-standard channel extraction scheme described in Example II is implemented on Xilinx Virtex 2v3000ff1152-4 FPGA associated with the dual DSP-FPGA Signalmaster kit provided by Lyrtech [66]. The implementation is tested using real-time inputs. The objective is to test the functionality of the architecture by selecting the desired frequency band to pass through the masking filter. The design diagram for Example II taken directly from the Simulink is shown in Fig. 5.6. The design diagram shows three blocks, the Lyrtech signalmaster controller, input subsystem and Xilinx ‘hwcosim’ block along with the Simulink representation of the architecture.

Figure 5.6. Simulink diagram for the CDMA/WCDMA channelizer using Lyrtech Signalmaster.

Lyrtech signalmaster controller consists of three components:-

- (1) The board configuration for configuring the FPGA/DSP and for downloading the bit-stream to FPGA.
- (2) Xilinx's system generator for generating the bit stream to be downloaded to the FPGA.
- (3) Log viewer which gives implementation information about the area and delay of the architecture used.

Input subsystem provides the real time input for the proposed architecture. A model based design using Matlab's Simulink and Xilinx's System generator was employed for the implementation purpose. System generator provides a hardware co-simulation interface that makes it possible to directly incorporate design running on signalmaster FPGA in a simulink simulation. The bit-stream of the proposed simulation architecture was generated using the Xilinx system generator. The generated bitstream can be downloaded to FPGA using board configuration block and it appears as a 'Xilinx hwcosim block'. The performances of the bitstream and the simulation architecture were checked to ensure that they are identical. The power consumption figures were calculated using X-power.

Table 5.2

Implementation results of multi-band filter and masking channel filter.

Standard	Total gate count	Total power (mW)	Total delay (ns)
TDMA/PDC 2-mode filter (Proposed)	251,422	396.5	7.316
TDMA/PDC (PC approach)	322,523	431.2	7.21
CDMA/WCDMA 2-mode filter (Proposed)	270,257	415.1	7.447
CDMA/WCDMA (PC approach)	402,899	479.5	7.241

Table 5.2 shows the implementation results. The proposed method is compared with the PC approach as PC approach is the only known method that can handle signals of multiple standards using dedicated channel filters. The proposed method offers an average gate count reduction of 26.5% over the PC approach for the above examples. The average reduction of power consumption achieved using the proposed method over the PC approach is 12.2%. The proposed multi-band and masking architecture has a slight increase in delay over the PC architecture. This overhead is due to the use of masking

filters at the second stage. It should be noted that the delay overhead in the proposed method is only about 2%, which is proportional to the length of the masking filter.

5.3 Practical Design Considerations of the Proposed Multi-band Filter and Masking Method

In this section, certain practical design considerations of the two-stage filter architecture are presented along with its trade-off with the PC architecture. As explained in Section 5.1, the complexity of the proposed architecture is determined by the total length of the multi-band and the masking filters. The total length of filters in the method is denoted as N_{tpa} . In many practical spectral scenarios, the inter-channel distance (frequency separation between adjacent channels) is sufficiently large so that the length of the masking filters is substantially less than that of the multi-band filter as the masking filters have wide transition-bands. Thus N_{tpa} would be shorter than the total length of filters in their PC counterpart, N_{ipc} . Therefore, in general, the complexity of proposed architecture is less than that of the PC architecture. However, if two adjacent channels are located closer, the inter-channel distance becomes shorter. Consequently, a sharp transition-band masking filter would be required to mask the unwanted channel, which in turn would increase the length of the masking filter. Thus, when the inter-channel distance is shorter than certain value, $\Delta f_{critical}$, then N_{tpa} will exceed N_{ipc} . The proposed two-stage architecture would not be a cost-effective solution in such a scenario. A theoretical framework for computing $\Delta f_{critical}$ is formulated here, in order to find out the cases where the proposed method offers a low complexity solution compared to the PC approach.

When the distance between any two adjacent stopbands of a multi-band filter, Δf_{ab} , is equal to $\Delta f_{critical}$, the length of the masking filter will be same as that of the multi-band filter. When $\Delta f_{ab} > \Delta f_{critical}$, then $N_{tpa} < N_{ipc}$, and the proposed architecture will have less complexity than the PC architecture. This can be illustrated with the help of Fig. 5.2. Fig. 5.2(a) shows the magnitude response of a three-band filter with passband widths BW_1 , BW_2 and BW_3 and transition-band widths of $TBW_1 = f_{s1} - f_{p1}$, $TBW_2 = f_{p2} - f_{s2}$ or $f_{s3} - f_{p3}$ and $TBW_3 = f_{p4} - f_{s4}$ or $f_{s5} - f_{p5}$ respectively. For extracting a desired channel, the unwanted channels are masked using masking filters of wider transition-band widths. The transition-

band widths of the masking filters for the first, second and third channel are defined as $MTBW_1 = f_{s2} - f_{p1}$, $MTBW_2 = f_{p2} - f_{s1}$, or $f_{s4} - f_{p3}$, $MTBW_3 = f_{p4} - f_{s3}$ respectively. As the stopband edges of two adjacent bands of the multi-band filter, f_{s1} and f_{s2} , becomes closer, i.e., when $TBW_1 = f_{s1} - f_{p1}$ is equal to $MTBW_1 = f_{s2} - f_{p1}$, their transition-band widths will become identical. At this condition, the multi-band filter has the maximum number of passbands (frequency channels) and the lengths of the multi-band filter and the masking filter will become identical. This scenario is called as the zero inter-channel distance (ZID) scenario. The ZID scenario in turn fixes the maximum number of channels in a multi-band filter.

Fig. 5.7 shows the multi-band frequency response of a three-band filter explained earlier with total given bandwidth, B_T , and total occupied bandwidth, B_0 , given by:

$$B_T = \sum_{i=1}^n (BW_i + D_i) + \sum_{i=2}^n 2 \times TBW_i + TBW_1 \quad (5.2)$$

$$B_0 = \sum_{i=1}^n BW_i + \sum_{i=2}^n 2 \times TBW_i + TBW_1 \quad (5.3)$$

where BW_i is the bandwidth of the i -th channel, D_i is the distance between the stopbands of i -th and $(i+1)$ -th bands, TBW_i is the transition-band width of i -th band, and n is the number of passbands (channels).

Figure 5.7 Frequency response of multi-band filter.

The length of the multi-band filter, N_{mb} , is fixed according to the most stringent transition-band width specification among the various passbands of the multi-band filter. N_{mb} can be evaluated using (5.4) [65]:

$$N_{mb} = \max_{i=1} \left(\frac{-2 \log_{10} (10 \times \delta_p \delta_s)}{3(TBW_i)/(2 \times B_T)} - 1 \right)^n \quad (5.4)$$

where δ_p is the peak passband ripple (PPR), δ_s is the peak stopband ripple (PSR) and n is the number of passbands in the multi-band filter. The length of the multi-band filter is determined based on the shortest transition-band width among n bands. The transition-band width of the masking filter used for extracting the i -th channel is given by

$$MTBW_i = (TBW_i + D_i) \quad (5.5)$$

Note that, if $MTBW_i$ is larger than the transition-band widths of individual bands in the multi-band filter by the inter-band distance D_i , the length of the masking filter will be shorter than that of the multi-band filter. The total length of the masking filters, N_{mask} , can be evaluated as

$$N_{mask} = \sum_{i=1}^n \left(\frac{-2 \log_{10} (10 \times \delta_p \delta_s)}{3(MTBW_i)/(2 \times B_T)} - 1 \right) \quad (5.6)$$

At ZID scenario of the multi-band filter, the distance between two adjacent stopbands is zero, i.e., $D_i = 0$. Hence, for the ZID scenario, $MTBW_i = (TBW)_i$, implies that the complexity of individual masking filters becomes equal to that of the multi-band filter. A comparison of the complexities of the proposed multi-band and masking architecture and the PC architecture corresponding to the ZID scenario is as follows. For simplicity, the transition-band widths of all the three bands in the multi-band filter shown in Fig. 5.7 are assumed to be identical. In PC approach, each channel is extracted using a distinct filter of length N ; thus three separate filters are needed. The total filter length using PC architecture, N_{pc} , is $3N$. On the other hand, the total filter length needed, N_{pa} , is $4N$ (N taps for the multi-band filter and $3N$ taps for the three masking filters), which is greater than the total filter length in PC architecture. Hence there is a need to find out the minimum distance between the stopband edges of the multi-band filter, $\Delta f_{critical}$, for which the proposed architecture offers a low complexity solution than the PC approach.

Assume N_{CH-1} , N_{CH-2} and N_{CH-3} are the lengths of the filters used in PC architecture to extract *CH-1*, *CH-2* and *CH-3* respectively. The complexity (determined by the total

filter length) of the proposed architecture will be less than that of the PC architecture if following condition is satisfied:

$$N_{mb} + N_{mask} \leq N_{CH-1} + N_{CH-2} + N_{CH-3} \quad (5.7)$$

Substituting the expressions for N_{mb} and N_{mask} from (5.4) and (5.6), equation (5.7) will become:

$$\max \left(\frac{-2 \log_{10}(10 \times \delta_p \delta_s)}{3(TBW_i)/(2 \times B_T)} - 1 \right)^3 + \sum_{i=1}^3 \left(\frac{-2 \log_{10}(10 \times \delta_p \delta_s)}{3(MTBW_i)/(2 \times B_T)} - 1 \right) \leq \sum_{i=1}^3 \left(\frac{-2 \log_{10}(10 \times \delta_p \delta_s)}{3(TBW_i)/(2 \times B_T)} - 1 \right) \quad (5.8)$$

Rearranging (5.8) and substituting value for $MTBW_i = TBW_i + D_i$, (5.9) is obtained. Note that the '-1' term in the above masking filter complexity equation is neglected in (5.9) for simplification.

$$\cong \sum_{i=1}^3 \left(\frac{1}{(TBW_i + D_i)} \right) \leq \frac{\sum_{i=1}^3 \left(\frac{-2 \log_{10}(10 \times \delta_p \delta_s)}{3(TBW_i)/(2 \times B_T)} \right) - \max \left(\frac{-2 \log_{10}(10 \times \delta_p \delta_s)}{3(TBW_i)/(2 \times B_T)} \right)^3_{i=1}}{-4/3 \times B_T \times \log_{10}(10 \times \delta_p \delta_s)} \quad (5.9)$$

The minimum D_i which satisfies (5.9) is $\Delta f_{critical}$, and for any D_i greater than $\Delta f_{critical}$, the complexity of the architecture is lower than that of PC method. For an n -band filter, the criterion for $\Delta f_{critical}$ given by (5.9) can be generalized as follows:

$$\sum_{i=1}^n \left(\frac{1}{(TBW_i + D_i)} \right) = \frac{\sum_{i=1}^n \left(\frac{-2 \log_{10}(10 \times \delta_p \delta_s)}{3(TBW_i)/(2 \times B_T)} \right) - \max \left(\frac{-2 \log_{10}(10 \times \delta_p \delta_s)}{3(TBW_i)/(2 \times B_T)} \right)^n_{i=1}}{-4/3 \times B_T \times \log_{10}(10 \times \delta_p \delta_s)} \quad (5.10)$$

For a given design problem of an n -mode channelizer, the multi-band and masking method offers a low complexity solution compared to the PC architecture if $N_{tpa} < N_{tpc}$, i.e.,

$N_{mb} + N_{mask} < \sum_{i=1}^n N_{CH-i}$, where N_{mb} is the length of the multi-band filter, N_{mask} is the total length of the masking filters, and the summation term is the total length of n filters in PC approach.

A complexity comparison of the proposed architecture with the PC architecture using the design example of a three-mode transceiver is presented here. A Personal Digital Cellular (PDC)/Digital Advanced Mobile Phone System (D-AMPS)/Personal

Handy-phone System (PHS) transceiver is used in this example: B_T is 400 kHz; sampling frequency, F_{ss} is 800 kHz; δ_p is 0.1 dB and δ_s is -48 dB. The three bands are assumed to be located at frequencies corresponding to the ZID scenario, i.e., $f_{s1} = f_{s2} = 30$ kHz and $f_{s3} = f_{s4} = 70$ kHz as shown in Fig. 5.8.

Figure 5.8 Magnitude response of PDC/D-AMPS/PHS channel filter.

Since $N_{tpa} > N_{tpc}$ for the ZID scenario (i.e., $D_i = 0$), the complexity of the multi-band and masking architecture is more than the PC architecture. The synthesis of above design example using Xilinx Virtex 2v3000ff1152-4 FPGA is done to obtain the complexities of these two methods. The gate count for channel filter implementations corresponding to the inter-channel distance is shown in Fig. 5.9. Obviously, for ZID scenario, the PC approach has less gate count requirement compared to the proposed architecture. When the inter-channel distance, D_i , increases, the transition-band widths of the masking filters will increase; the lengths of the masking filters will decrease, and consequently the overall complexity of the architecture will reduce. When $D_i \geq \Delta f_{critical}$, the multi-band and masking architecture outperforms PC architecture in terms of complexity. $\Delta f_{critical}$ computed using (5.10) is 2.5 kHz as indicated in Fig. 5.9. Note that $\Delta f_{critical} = 2.5$ kHz is only a small fraction of the bandwidths of PDC (25 kHz), D-AMPS (30 kHz) and PHS (300 kHz). For a given total bandwidth B_T , channel occupancy (capacity) is maximum when $B_0/B_T = 1$, which is a feasible scenario in PC approach. A large $\Delta f_{critical}$ would reduce B_0 which will in turn impose a limitation to the number of channels that can be accommodated for a given B_T . As $\Delta f_{critical}$ is small in the proposed method, it has only very small impact on the maximum number of channels that can be accommodated for a given total bandwidth B_T , i.e., B_0/B_T is only slightly less than unity.

Figure 5.9 Complexity analysis of PC architecture and the proposed architecture for varying inter-channel distance (D_i).

In conclusion, the proposed multi-band and masking approach will result in a less complex architecture than PC architecture in most practical situations on account of its small $\Delta f_{critical}$ and is only slightly inferior to PC architecture in terms of maximum channel capacity. Reconfigurability is achieved by using suitable masking filters at the back end. Even though the proposed method reduces complexity when compared to PC approach, it has to be acknowledged that the multi-band front end filter still needs a large filter order. Hence there is a need to look into unconventional reconfigurable filter design techniques so as to further reduce the complexity.

This requirement motivated the idea proposed in Section 5. 4 :- a channel filtering and filter bank based on Multi-stage FRM (MSFRM) technique. The basic idea was to compose the over-all sharp transition-band filter using three wide transition-band subfilters. Since the subfilters have wide transition-band specifications, they can be of lower-order. Consequently, the over-all complexity of FRM-based filter will be much less than sharp transition-band FIR filters designed using conventional methods. The proposed architecture is unique in a way that it is able to effectively exploit the redundancy in multi-stage realization by utilizing the common masking filters thus achieving very low complexity. The proposed reconfigurable filter bank overcomes certain shortcomings of

existing filter banks such as simultaneous extraction of non-uniform bandwidth channels, extremely narrowband channels and multi-mode operation. The proposed MSFRM filter and filter bank have the flexibility of changing frequency responses dynamically with very low hardware overhead.

5.4 Multi-Stage Frequency Response Masking (MSFRM) Channel Filter

A low complexity reconfigurable channel filter architecture is presented in this section, which reuses the same hardware for extracting non-uniform bandwidth channels. The filters in SDR channelizers should have sharp transition-band to meet the stringent adjacent channel attenuation specifications of wireless communication standards. Therefore, longer length FIR filters are needed for realizing channel filter. As the filter length increases, its complexity also increases. FRM [63] technique is an approach to build sharp transition-band filters with reduced complexity.

5.4.1 Review of Frequency Response Masking

The FRM technique is one of the most efficient techniques to realize FIR filters with sharp transition-bands, using several wide transition-band subfilters [63]. The advantage of FRM technique is that, the resulting filter will have many sparse coefficients and hence will have low complexity. FRM filter design procedure starts with the design of a prototype lowpass filter $F_a(z)$ of odd length N , also known as the *modal* filter. The complementary of the modal filter $F_c(z)$ can be expressed as

$$F_c(z) = z^{-\frac{(N-1)}{2}} - F_a(z) \quad (5.11)$$

Replacing each delay elements of both filters by L delays, two filters with transfer functions $F_a(z^L)$ and $F_c(z^L)$ are formed. The transition-band widths of $F_a(z^L)$ and $F_c(z^L) = z^{-(L(N-1))/2} - F_a(z^L)$ are a factor of L narrower than that of $F_a(z)$. In the FRM technique, two filters $F_{MA}(z)$ and $F_{MC}(z)$, are cascaded to $F_a(z^L)$ and $F_c(z^L)$, respectively as shown in Fig. 5.10. The transfer function of the overall filter is given by [63]:

$$F(z) = F_a(z^L)F_{MA}(z) + [z^{-\frac{L(N-1)}{2}} - F_a(z^L)]F_{MC}(z) \quad (5.12)$$

Figure 5.10 FIR filter architecture based on FRM technique [63].

Note that the group delay of the filters $F_{MA}(z)$ and $F_{MC}(z)$ must be equal, and $L(N-1)$ in equation (5.12) must be an even number. If they are of unequal length, zero valued coefficients must be padded to the shorter one.

The frequency illustration of modal filter and its complementary filter $F_c(z)$ is shown in Fig. 5.11(a) and 5.11(b) respectively. Replacing each delay elements by L delays, filters with transfer function $F_a(z^L)$ and $F_c(z^L)$ are formed as shown by the solid and the dotted line in Fig. 5.11(c) respectively. The transition-band width and the bandwidth of $F_a(z^L)$ is a factor L narrower than that of $F_a(z)$. Two masking filters, $F_{MA}(z)$ and $F_{MC}(z)$, shown in Fig. 5.11(d) are used to remove the undesired frequency components from the output of the modal filter and their outputs are summed to result the sharp transition response as shown in Fig. 5.11(e).

Figure 5.11 Frequency response illustration of FRM [63].

The design steps for the sub-filters involve the solution of the expressions [63]:

$$m = \lfloor f_p L \rfloor, f_{ap} = f_p L - m, f_{as} = f_s L - m, \quad (5.13)$$

$$f_{map} = f_p, f_{mas} = \frac{m+1-f_{as}}{L}, f_{mcp} = \frac{m-f_{ap}}{L}, f_{mcs} = f_s \quad (5.14)$$

where $\lfloor x \rfloor$ denotes the largest integer less than x , L is the interpolation factor (same as the number of delays) for $F_a(z)$, f_p and f_s are the passband and stopband edges of the overall filter, f_{ap} and f_{as} are the passband and stopband edges of the modal filter ($F_a(z)$), f_{map} and f_{mcp} are the passband edges and f_{mas} and f_{mcs} are the stopband edges of the two masking filters respectively. By suitable selection of the passband and stopband edges of the modal filter and the masking filters, any sharp transition-band FIR filter can be realized. Low complexity realization is possible in FRM design technique due to the shorter length of modal and masking filters.

The FRM technique was originally proposed as a low complexity solution for the design of sharp transition-band FIR filters [63]. It has been reported in [67] that FRM based filter can be exploited to integrate architectural reconfigurability, which can be well explained in using the expressions (5.13) [67]. For simplicity, a dual-mode channelizer is considered. Let f_{p1} and f_{s1} represent the passband and the stopband frequencies of the channel filter corresponding to one of the communication standards (modes), and f_{p2} and f_{s2} , the respective frequencies of the other standard. Reconfigurability can be achieved by using the same subfilters shown in Fig. 5.10 for both the standards. The parameters f_{ap} and f_{as} remain unchanged for both the standards, i.e., the same modal filter is employed for both the standards and the masking filters can be reconfigured by swapping the LUT values. Thus,

$$f_{p1}L_1 - \lfloor f_{p1}L_1 \rfloor = f_{p2}L_2 - \lfloor f_{p2}L_2 \rfloor \quad (5.15)$$

$$f_{s1}L_1 - \lfloor f_{s1}L_1 \rfloor = f_{s2}L_2 - \lfloor f_{s2}L_2 \rfloor \quad (5.16)$$

where L_1 and L_2 denote the up-sampling factor for the two standards which can be obtained by solving (5.15) and (5.16). Thus by changing the number of delay elements, it is possible to reconfigure the same modal filter to work for both the standards. The dual-

mode channelizer can be extended to incorporate additional communication modes by choosing appropriate number of delay elements. For example, if a third standard needs to be incorporated, then the number of delays (Interpolation factor, L) can be obtained by substituting the filter specification corresponding to the third standard in expressions (5.15) and (5.16).

In order to realize an arbitrary bandwidth channel by keeping the modal filter specifications a constant, the value of L may need to vary from very low value to very high values. Increasing the value of L will produce two effects: It enables us to have more flexibility with the bandwidth, but increases the complexity of masking filters. The output of the modal filter when interpolated by L results in $(L+1)$ multi-band outputs which lies close together. As L is increased, the multiple passbands of the multi-band response will be located closer to each other. Consequently, very narrow transition-band masking filter would be required to isolate the band of interest from the multi-band response. As the complexity of the masking filters increases when the transition-band becomes narrow, the overall filter complexity also increases. The above problems of a Single-Stage FRM (SSFRM) proposed in [67] can be overcome by employing the multi-stage FRM (MSFRM) technique presented in this section. Fig. 5.12 shows the representation of a two-stage FRM.

Figure 5.12 Architecture of proposed two-stage FRM.

Two-stage FRM is realized using a SSFRM along with the FRM implementation of modal filter. $F_a^{(1)}[z^{L_1}]$ is the modal filter for the first stage interpolated by L_1 . The modal filter of the first stage can be realized using FRM technique in the second stage. Overall transfer function for a 2-stage FRM can be realized as:

$$F_a^{(0)}(z) = F_a^{(1)}(z^{L_1})F_{MA}^{(1)}(z) + F_c^{(1)}(z^{L_1})F_{MC}^{(1)}(z) \quad (5.17)$$

where $F_a^{(1)}(z^{L_1})$ is the modal filter for first stage, $F_{MA}^{(1)}(z)$ and $F_{MC}^{(1)}(z)$ are the masking filters of the first stage.

Thus two-stage FRM allows low complexity realization of sharp transition-band filters which has large L values. In this proposed work, reconfigurability is integrated with the multi-stage FRM architecture. This enables the channel filter to be reconfigured for a number of wireless radio channels of multiple standards. In the next section, detailed design steps of the proposed multi-stage channel filter are provided.

5.4.2 Design Steps of Proposed Multi-stage Channel Filter

1. The passband and stopband of the modal filter is fixed initially as f_{ap} and f_{as} .
2. Let f_p and f_s are the passband and stopband edges of the overall channel filter to be designed and let L the number of delays for modal filter $F_a(z)$ to realize the overall channel filter.
3. Calculate the filter length of modal filter, (N_{modal}) using (5.1) [65],

$$N_{modal} = \frac{-2 \log_{10} (10 \times \partial_1 \partial_2)}{3(f_{as} - f_{ap})} - 1, \quad (5.18)$$

where ∂_1 and ∂_2 are the peak passband and stopband ripples respectively, and $(f_{as}-f_{ap})$ is the normalized bandwidth and rounded to the nearest odd integer value. Also calculate length of the masking filter, $N_{masking}$, depending upon the number of interpolated channels of the modal filter to be masked for the given L value. If $N_{masking} > N_{modal}$, then factorize L into two factors, $L=L_i.L_j$. Check whether $N_{masking} < N_{modal}$ for the all the factorized L values. If the condition is not satisfied proceed further factorizing L_i or L_j until $N_{masking} < N_{modal}$ or until further factorization cannot be done.

4. Let L_i ($i = 1$ to k), where k is the last factor, be the factorized values for given L . Let L_1 be the first stage interpolation factor and the design of the first stage masking filter can be done as follows. First, find out the value of f_p and f_s of the intermediate stage for the corresponding L_1 from the equations (5.13). Using equation (5.14) the specifications of the masking filter for the given f_p and f_s can be found out.
5. The next stage masking filter can be designed by using the f_p and f_s values with respect to L_2 and by repeating the step 4 and by replacing L_1 with L_2 .
6. Proceed with the design of the masking filters till the last factor of L .

The proposed architecture has more flexibility when compared to SSFRM, as it is able to achieve multiple levels of reconfigurability depending on the number of stages in FRM architecture. The advantages of the proposed multi-stage FRM channel filter over [67] are:

- a. The channel filter for a desired non-uniform bandwidth can be designed using the same modal filter. For that, wide ranges of L values are needed. As explained in Section 5.4.1, the complexity of the masking filter increases as L increases, which will in turn impose a limitation in selecting large values of L . Using MSFRM method, the complexity can be reduced by factorizing the L value and implementing them in multiple stages. The complexity of the masking filters reduces as they are designed for a lower L value for the same modal filter. Hence filter level reconfigurability is possible by dynamically changing the number of delays.
- b. For a multi-standard channel filter, the redundant computations in multiple stages can be eliminated by performing those computations only once and then reusing them in other stages. This architectural property results in an enormous savings in terms of area complexity and power. The redundancy can be exploited if any of the factorized L values (i.e. L_1 or L_2 for the two-stage FRM example illustrated earlier) for various standards have common factors. If L_1 is same for two standards then they can use the same first stage masking filters as

the modal filter does not change. The same criteria will apply for all the stages of multi-stage channel filter.

5.5 An Example of MSFRM-Based Channel Filter Architecture

The proposed architecture of the channel filter is shown in Fig. 5.13, where $F_a(z)$ represents modal filter, CD_1 and CD_2 represents the complementary delay filters. $F_{MA1}(z)$, $F_{MA2}(z)$, $F_{MA3}(z)$, $F_{MC2}(z)$ and $F_{MC3}(z)$ represents the first stage masking filters and $F_{MA11}(z)$, $F_{MA12}(z)$ and $F_{MA13}(z)$ are the second stage masking filters. A complementary filter can be realized by subtracting the output of the $L+1$ band filter from a suitably delayed version of the input.

Figure 5.13 Proposed MSFRM based channel filter architecture.

In this illustrative design example, the specifications of the modal filter are chosen as follows. Passband edge, $f_{ap} = 0.2$ and stopband edge, $f_{as} = 0.3$. The design of a 12-mode channel filter with varying normalized bandwidth (BW) ranging from 0.025 to 0.4 is considered here. An n -mode channel filter can be reconfigured to obtain n distinct bandwidth channels corresponding to n communication standards. The channels of interest are given in Table 5.3 along with the values of L for realizing the channels. The number of stages of implementation is determined by the number of factors of L , and each factor satisfies the condition in the design flow explained in Section 5.4.2. The factorized L values are also shown in Table 5.3. The passband and stopband edge information of all the masking filters and channels of interest are indicated in Fig. 5.13. The final set of interpolation values are indicated alongside the data path at the inputs of the four multiplexers labeled ‘3’ in Fig. 5.13.

Table 5.3

Details of the number of channels extracted using proposed MSFRM based architecture.

Channels of interest	Interpolation factors	No of stages of Implementation	L values of different stages of FRM
$Y_1=[f_{p1}=0.1, f_{s1}=0.15]$	$L_1=2$	1	
$Y_2=[f_{p2}=0.05, f_{s2}=0.075]$	$L_2=4$	2	$L_{2_1}=2, L_{2_2}=2$
$Y_3=[f_{p3}=0.034, f_{s3}=0.05]$	$L_3=6$	2	$L_{3_1}=2, L_{3_2}=3$
$Y_4=[f_{p4}=0.025, f_{s4}=0.0375]$	$L_4=8$	2	$L_{4_1}=2, L_{4_2}=4$
$Y_5=[f_{p5}=0.4, f_{s5}=0.433]$	$L_5=3$	1	
$Y_6=[f_{p6}=0.2, f_{s6}=0.217]$	$L_6=6$	2	$L_{6_1}=3, L_{6_2}=2$
$Y_7=[f_{p7}=0.133, f_{s7}=0.144]$	$L_7=9$	2	$L_{7_1}=3, L_{7_2}=3$
$Y_8=[f_{p8}=0.1, f_{s8}=0.108,]$	$L_8=12$	2	$L_{8_1}=3, L_{8_2}=4$
$Y_9=[f_{p9}=0.24, f_{s9}=0.26]$	$L_9=5$	1	
$Y_{10}=[f_{p10}=0.12, f_{s10}=0.13]$	$L_{10}=10$	2	$L_{10_1}=5, L_{10_2}=2$
$Y_{11}=[f_{p11}=0.08, f_{s11}=0.0866]$	$L_{11}=15$	2	$L_{11_1}=5, L_{11_2}=3$
$Y_{12}=[f_{p12}=0.06, f_{s12}=0.065]$	$L_{12}=20$	2	$L_{12_1}=5, L_{12_2}=4$

From the right-most column in Table 5.3, it is noted that the first stage interpolation values are identical for channels Y_1 to Y_4 (i.e. $L_{2_1} = L_{3_1} = L_{4_1} = 2$), and same is the case for channels Y_5 to Y_8 (identical interpolation value of 3) as well as Y_9 to

Y_{12} (identical interpolation value of 5). The above redundancy in the architecture can be exploited using a single first stage masking filter for extracting channels Y_1 to Y_4 . The same is applicable for channels Y_5 to Y_8 and Y_9 to Y_{12} . Hence in total, only three first stage masking filters and two complementary first stage masking filters ($F_{MA1}(z)$, $F_{MA2}(z)$, $F_{MA3}(z)$, $F_{MC2}(z)$ and $F_{MC3}(z)$) are needed as shown in Fig. 5.3.

The redundancy of the architecture can be further exploited using the second stage factorized L values. It can be noted that for channels Y_2 , Y_6 , Y_{10} the second stage L value is 2 (i.e. $L_{2_2} = L_{6_2} = L_{10_2} = 2$), and for Y_3 , Y_7 , Y_{11} the second stage L value is 3 and for Y_4 , Y_8 , Y_{12} the second stage L value is 4. Hence only three second stage masking filters, indicated as $F_{MA11}(z)$, $F_{MA12}(z)$ and $F_{MA13}(z)$ in Fig. 5.13, are needed.

The architecture of the modal filter is shown in Fig. 5.14 (a) and that of the first set of masking filters ($F_{MA1}(z)$, $F_{MA2}(z)$, $F_{MA3}(z)$, $F_{MC2}(z)$ and $F_{MC3}(z)$) is shown in Fig. 5.14(b). The delay elements for the modal filter are selected by the multiplexers shown as Mux-1 using mode select 1 as shown in Fig. 5.14(a). The modal filter outputs are masked using multi-stage masking technique so as to reduce the overall complexity.

The F_{MA1} shown in Fig. 5.13 is the shared first stage masking filter for the output (for delay elements 2, 4, 6 and 8) of the modal filter. Each delay element in the masking filter is also replaced by K delay elements ($K=1, 2, 3$ or 4) so as to get an output response interpolated by K . The $L+1$ bands of the interpolated modal filter are masked using the $K+1$ bands of the masking filter and thus $K+1$ bands are extracted. These $K+1$ bands are further masked to extract a single band, using second stage masking filters indicated as F_{MA11} , F_{MA12} and F_{MA13} in Fig. 5.13. It is noted from Fig. 5.13 that, using multi-stage masking the redundancy within the circuit can also be exploited. The multiplexer Mux-3 in Fig. 5.13 selects those bands which can be masked using the same second stage masking filter. For example, bands with interpolating values 4, 6 and 8 are masked using F_{MA11} . Mux-4 shown in Fig. 5.13 is used to select the channel-of-interest.

Figure 5.14(a) Modal filter architecture.

Figure 5.14(b) Architecture of first set masking filters (F_{MA1} , F_{MA2} , F_{MA3} , F_{MC2} , F_{MC3}).

The complementary delays can be obtained using the equation, $CD = (N_{modal} - 1) \times L/2$, where N_{modal} is the length of the modal filter. For illustration, let us take $\delta_1 = 0.1$ dB and $\delta_2 = -40$ dB for all the sub-filters in Fig. 5.13. The length of the modal filter (19 taps), is obtained from equation (5.18). The complementary delay values are calculated and the architecture of the complementary delays for F_{MC2} and F_{MC3} are shown in Figures 5.15(a) and 5.15(b) respectively. In order to select Y_5 as the output, the select signal for Mux-1 is “100” to select the L value as 3, “00” for Mux-2 to select complementary delay value as Z^{-27} , “01” for Mux-3 to select the second stage masking filter and “00” for Mux-4 for selecting the final output.

Figure 5.15(a) Complementary delay architecture for F_{MC2} .

Figure 5.15(b) Complementary delay architecture for F_{MC3} .

5.5.1 Comparison of the proposed MSFRM method with Single-Stage FRM (SSFRM)

The proposed MSFRM architecture is compared with the SSFRM architecture for the same set of channels. Fig. 5.16 shows the SSFRM architecture for the 12-mode channel filter. The modal filter for the single-stage and multi-stage channel filter remains the same. The channels-of-interest are indicated along the data path at the input of Mux-2 in Fig. 5.16. It can be noted that for SSFRM masking filters cannot be shared as they have different specification according to the bands extracted. The frequency edges of the masking filters (F_{MAS} and F_{MCS}) are given in Fig. 5.16 and the channel-of-interest is selected through Mux-2.

Figure 5.16 SSFRM based channel filter architecture.

Table 5.4 gives the filter length comparison of the filters in the proposed MSFRM based architecture shown in Fig. 5.13 and SSFRM based architecture for the 12-mode channel filter explained in Section 5.5. The lengths of all the filters are obtained from equation (5.18), with $\sigma_1=0.1$ dB and $\sigma_2=-40$ dB. The proposed MSFRM based channel filter enjoys sharing of first and second stage filters by exploiting the redundancy of the architecture. It can be noted from Table 5.4 that the second stage masking filters for channels Y_1 to Y_4 with length $\{7, 11$ and $15\}$ is repeated for channels Y_5 to Y_8 and also for Y_9 to Y_{12} . It can also be noted from Fig. 5.13 that channels Y_1 to Y_4 are generated using

the same first stage masking filter of length 7, with interpolation values (L) ranging from 1 to 4. The case repeats for channels Y_5 to Y_8 and for Y_9 to Y_{12} . Thus the total filter length for the proposed multi-stage 12-mode channel filter is 135, which is the sum of the length of modal filter (19 taps), first stage masking filters ($7+19+19+19+19 = 83$ taps), second stage masking filters ($7+11+15 = 33$ taps). The total filter length for the SSFRM based channel filter [67] shown in Fig. 5.16 is $731(= 19 + 7 + 15 + 23 + 31 + (19+19) + (23+23) + (35+35) + (47+47) + (19+19) + (39+39) + (59+59) + (77+77))$. Thus the complexity of proposed MSFRM based channel filter is only less than one-fifth of the complexity of the filter realized using SSFRM.

Table 5.4
Comparison of filter lengths in proposed MSFRM and SSFRM based channel filter.

Channels-of-interest	Filter length of modal filter	Filter length of first stage masking filter	Filter length of second stage masking filter	Filter length of masking filter for SSFRM
$Y_1=[f_{p1}=0.1, f_{s1}=0.15]$	19	7		7
$Y_2=[f_{p2}=0.05, f_{s2}=0.075,]$	19	7	7	15
$Y_3=[f_{p3}=0.034, f_{s3}=0.05]$	19	7	11	23
$Y_4=[f_{p4}=0.025, f_{s4}=0.0375]$	19	7	15	31
$Y_5=[f_{p5}=0.4, f_{s5}=0.433]$	19	19+19		19+19
$Y_6=[f_{p6}=0.2, f_{s6}=0.217]$	19	19+19	7	23+23
$Y_7=[f_{p7}=0.133, f_{s7}=0.144]$	19	19+19	11	35+35
$Y_8=[f_{p8}=0.1, f_{s8}=0.108]$	19	19+19	15	47+47
$Y_9=[f_{p9}=0.24, f_{s9}=0.26]$	19	19+19		19+19
$Y_{10}=[f_{p10}=0.12, f_{s10}=0.13]$	19	19+19	7	39+39
$Y_{11}=[f_{p11}=0.08, f_{s11}=0.0866]$	19	19+19	11	59+59
$Y_{12}=[f_{p12}=0.06, f_{s12}=0.065]$	19	19+19	15	77+77

5.5.2 Implementation Results

The proposed channel filter architecture based on MSFRM is implemented and compared with the SSFRM architecture for the 12-mode channel filter explained in section 5.5.1. The architecture was implemented on Xilinx Virtex 2v3000ff1152-4 FPGA associated with the dual DSP-FPGA Signalmaster kit provided by Lyrtech [66]. It was tested using

real-time Gaussian noise as the input signal. The basic objective is to test the functionality of the architecture by selecting the desired channel to pass through the masking filter. The design diagram for proposed MSFRM based channel filter architecture taken directly from the Simulink is shown in Fig. 5.17. The design diagram shows Lyrtech signalmaster controller blocks, Gaussian noise input and Xilinx hardware cosimulation (*hwcosim*) block along with the Simulink representation of the architecture.

Figure 5.17 Implementation diagram for the 12-mode MSFRM based channel filter architecture.

The bit-stream of the proposed architecture is generated using the Xilinx system generator. The bit-stream is downloaded to FPGA using board configuration block. The downloaded bit-stream is indicated as the ‘*hwcospim*’ block. The power is calculated using X-power. Table 5.5 shows the implementation results. The proposed MSFRM based method offers an average gate count reduction of 38.5% over [67] for the above example. The average reduction of power consumption achieved using the proposed method over [67] is 18.1%. Note that the proposed architecture has a slight overhead of delay (1.3%) when compared to [67]. Delay measurement given in Table 5.5 is the net combinatorial time for the first sample to appear at the output since the first sample is fed to the architecture. However, this delay overhead is negligible when compared with the savings in gate count and power consumption.

Table 5.5
Implementation results of MSFRM based channel filter.

Standard	Total equivalent gate count	Total power (mW)	Total delay (ns)
Proposed multi-stage architecture	677,091	417	15.405
Single-stage FRM channel filter [67]	1100,380	509	15.21

5.5.3 Design Examples

In this section, the proposed reconfigurable architecture for dual-standard Code Division Multiple Access (CDMA)/Wideband-CDMA (WCDMA) channel filter is presented. The dual-standard examples are taken here for convenience; the proposed multi-stage FRM can support more than two standards (as illustrated using the 12-mode channelizer in Section 5.5). The sampling rate chosen is 105 MHz. The channel filters can extract 1250 kHz CDMA channels and 5 MHz WCDMA channels from the input signal after down sampling by a factor of 5. The passband and stopband edges are 1250 kHz and 1300 kHz respectively for CDMA and 5000 kHz and 5050 kHz respectively for WCDMA. The peak passband ripple is chosen as 0.1 dB. The filter stopband specification is chosen as -40 dB for both the standards. Reconfigurability can be achieved as follows. The channel

specifications for CDMA is, $f_p = (1250 \times 5/105000) = 0.06$ and $f_s = (1300 \times 5/105000) = 0.062$ at the output where 150 kHz is the sampling frequency and the down sampling factor is 5. Now considering WCDMA, a channel with specifications $f_p = (5000 \times 5/105000) = 0.24$ and $f_s = (5050 \times 5/105000) = 0.242$ is needed at the output. The CDMA/WCDMA reconfigurable channel filter can be designed using the multi-stage example filter architecture explained in Section 5.5. The CDMA signals are extracted through channel $Y_{12} = [f_{p12}=0.06, f_{s12}=0.065]$ and WCDMA channels through $Y_9 = [f_{p9}=0.24, f_{s9}=0.26]$, shown in Table 5.4. The realization of CDMA/WCDMA channel filter using channels Y_{12}/Y_9 is shown with the help of frequency responses in Fig. 5.18. Figures 5.18(a), 5.18(b) and 5.18(c) show the CDMA extraction while Figures 5.18(d), 5.18(e) and 5.18(f) show the WCDMA extraction. Fig. 5.18(a) shows the interpolated modal filter response indicated as solid line and the first stage masking filter response indicated as dotted line. Fig. 5.18(b) shows the complementary response of the interpolated modal filter indicated as solid line and the first stage complementary masking filter response indicated as dotted line. The output responses of Figures 5.18(a) and 5.18(b) are added together to realize the response shown in Fig. 5.18(c) indicated by solid lines. The second stage masking filter response shown in Fig. 5.18c masks the unwanted bands thus helping to extract the CDMA channel.

Figure 5.18(a) CDMA interpolated modal filter response and first stage masking filter response.

Figure 5.18(b) CDMA interpolated complementary modal filter response and first stage complementary masking response.

Figure 5.18(c) CDMA addition result of the extracted output from figures (a) and (b) (which is the CDMA channel) and masked using second stage masking filter.

Fig. 5.18(d) shows the interpolated modal filter response indicated as solid line and the first stage masking filter response indicated as dotted line. Fig. 5.18(e) shows the complementary response of the interpolated modal filter indicated as solid line and the first stage complementary masking filter response indicated as dotted line. The output

responses of Fig. 5.18(d) and 5.18(e) are added together to realize the WCDMA channel shown in Fig. 5.18(f).

Figure 5.18(d) WCDMA interpolated modal filter response and first stage masking filter response.

Figure 5.18(e) WCDMA interpolated complementary modal filter response and first stage complementary masking response.

In the proposed MSFRM based method, the total filter length for CDMA/WCDMA extraction is 110, which is the sum of 19 ($= N_{modal}$), 19×2 ($= N_{mask}$ for CDMA) and $(19 \times 2) + 15$ ($= N_{mask}$ for WCDMA). Using SSFRM method, the total filter

length is 211. Thus the proposed MSFRM based channel filter offers 47.9 % reduction in complexity over SSFRM based channel filter.

Figure 5.18(f) WCDMA addition result of the extracted channels of figures (d) and (e) (which is WCDMA channel).

5.5.4 Functionality Test

The channelization functionality of the proposed MSFRM based channel filter architecture for the CDMA/WCDMA design example is tested as follows. The specifications of the channels for the extraction of CDMA/WDCMA are same as that of design example in Section 5.5.3. The simulation input signal is taken as a Gaussian white noise with a sampling rate of 105 MHz. Fig. 5.19(a) shows the frequency spectrum of the input signal represented as dotted line and the CDMA spectrum extracted through proposed channel filter architecture is represented as solid line. Fig. 5.19(b) shows the WCDMA output spectrum extracted using the proposed MSFRM based channel filter (solid line) along with the input spectrum (dotted line). The spectrum of CDMA satisfies the design specifications ($f_p=1250$ kHz and $f_s=1300$ kHz $\delta_p=0.1$ dB and $\delta_s=-40$ dB) and WCDMA satisfies the design specifications ($f_p=5000$ kHz, $f_s=5500$ kHz and $\delta_p=0.1$ dB and $\delta_s=-40$ dB) as shown in Figures 5.19(a) and 5.19(b). The average passband error values for CDMA and WCDMA signals extracted using proposed MSFRM based channel filter are 0.042 dB and 0.063 dB respectively. Thus the proposed MSFRM based

reconfigurable channel filter is capable of efficiently extracting desired channels of multiple communication standards.

Figure 5.19(a) Input signal spectrum and CDMA spectrum extracted using proposed MSFRM based channel filter.

Figure 5.19(b) Input signal spectrum and WCDMA spectrum extracted using proposed MSFRM based channel filter.

5.6 MSFRM-Based Filter Bank Architecture for SDR

The MSFRM approach explained in previous sections can be used for filter bank implementation. However in the case of filter banks, the multiple stages of masking filters cannot be reused. For a MSFRM based filter bank implementation, masking filter redundancy cannot be exploited as the masking filters have to be replicated for simultaneous extraction of different channels of wireless radio signals. It can be noted that MSFRM based filter bank is less complex than that of SSFRM based filter bank. This is be verified using the same 12 channel variable filter bank implementation explained earlier. The architecture for a multi-stage implementation of 12 channel variable filter bank is shown in Fig. 5.20. The interpolation (L) values for each output are indicated inside the rectangular block representing the modal filter. The values signify the interpolation factors of the respective outputs. The modal filter architecture is similar to Fig. 5.14(a) excluding for the output multiplexer so as to obtain multiple outputs simultaneously.

The first 4 bands are extracted without adding the complementary delayed response of the modal filter. But the rest of the outputs are taken after adding the interpolated complementary response of the modal filter. The lengths of all the filters are obtained from equation (5.18), with $\partial_1=0.1$ dB and $\partial_2=-40$ dB. The complementary delays (CDs) can be obtained using the equation, $CD = (N_{modal}-1) * L/2$, where N_{modal} is 19. The values of CD_5 to CD_{12} are 27, 54, 81, 108, 27, 54, 81 and 108 respectively. The passband and stopband edges of each masking filter are also shown inside the square brackets in respective filter blocks in Fig. 5.20. The architectures of the first stage masking filters are same as that in Fig. 5.14(b). The delays of first stage masking filters are also replaced by K delays (where $K= 1, 2, 3, 4$) for F_{MA1-4} , F_{MA5-8} , F_{MA9-12} , F_{MC5-8} and F_{MC9-12} (as indicated in the right side of the rectangular blocks representing the filters). The second stage masking filters F_{MA2_11-13} , F_{MA2_21-23} and F_{MA2_31-33} are also shown in Fig. 5.20 along with its passband and stopband edges. The extracted channels are indicated in Fig. 5.20 using the channel variables Y_1 to Y_{12} and the passband and

stopband edges of each channel are represented inside the square brackets at the output of the proposed MSFRM based filter bank architecture.

Figure 5.20 Proposed MSFRM based filter bank architecture.

The proposed MSFRM based filter bank architecture is compared with SSFRM based filter bank for the same set of outputs as indicated in Fig. 5.21. For the SSFRM

based filter bank, the total filter length will be same as that of the channel filter explained in the earlier example, i.e.=663, whereas the total filter length of MSFRM based filter bank architecture is, $\{\{\text{Length of modal filter}\} + \{\text{Total length of first stage masking filters}\} + \{\text{Total length of second stage masking filters}\}\} = \{\{19\} + \{7 \times 4 + (19+19) \times 4 + (15+15) \times 4\} + \{7 + 11 + 15 + 7 + 11 + 15 + 7 + 11 + 15\}\} = 418$. Thus the total complexity reduction of the proposed MSFRM based filter bank method compared to that of SSFRM based filter bank is 36.95 %.

Figure 5.21 SSFRM based filter bank architecture [67].

5.6.1 Quantitative Comparison of MSFRM based Channelizer with other Methods

The proposed MSFRM based filter bank architecture is compared in terms of multiplication rate with that of the PC approach, DFTFB, MPRFB [26], GFB [17] and SSFRM based filter bank [67]. The comparison is shown in Table 5.6. Multiplication rate of a channelizer is defined as the total number of multiplications per sampling period for extracting N_I number of channels simultaneously. The sampling frequency of the signal at the input of the filter bank is f_s . The value of D represents the downsampling factor. The multiplications in a channelizer comprises of: (1) Channel filtering multiplications, (2) Digital down conversion multiplications, (3) Multiplications for modulation of filters.

Table 5.6
Quantitative comparison of multiplication rate of a channelizer.

	PC approach	DFTFB [17]	MPRFB [26]	GFB [17]	SSFRM filter bank [67]	Proposed MSFRM filter bank
Filter	$\frac{2N_I}{D} \cdot P \cdot f_s$	$P \cdot f_s$	$2P \cdot f_s$	$P \cdot f_s$	$l \cdot f_s$	$n \cdot f_s$
Digital Down Conversion	$2N_I \cdot P \cdot f_s$	-	-	-		
Modulation of filters	-	$\frac{\log_2 D}{2} \cdot f_s$	$\log_2 D \cdot f_s$	$N_I \cdot f_s$		
Total	$\frac{2N_I}{D} \cdot P \cdot f_s + 2N_I \cdot P \cdot f_s$	$P \cdot f_s + \frac{\log_2 D}{2} \cdot f_s$	$2P \cdot f_s + \log_2 D \cdot f_s$	$P \cdot f_s + N_I \cdot f_s$	$l \cdot f_s$	$n \cdot f_s$

In Table 5.6, P represents the number of nonzero filter coefficients of the prototype filter for the PC approach, DFTFB, MPRFB and GFB whereas ‘ l ’ represents the total number of nonzero coefficients of the modal filter and masking filter for the SSFRM filter bank and ‘ n ’ represents the total number of nonzero coefficients of the modal filter and different stages of masking filters for the proposed MSFRM based filter bank. The multiplication rate values for PC approach, DFTFB and GFB are taken from [17]. The multiplication rate for the MPRFB [26] is calculated based on the multiplication rate of DFTFB in [17]. As the analysis and synthesis need two prototype filters, the

multiplications for channel filtering are doubled i.e. $2P \cdot f_s$ for [26]. The MPRFB also needs both DFT and IDFT and hence the multiplications involved in modulation of filters are also doubled compared to that of DFT filter banks. For SSFRM based filter bank, the multiplication rate can be calculated as the product of the total number of nonzero coefficients for the modal filter and the masking filters, multiplied with sampling frequency. As all the outputs shown in the example above occurs in the base band, there is no need for DDC or modulation of filters. Hence FRM methods results in a low complex solution for limited number of channels. The value of ' l ' is very much less than that of P as the entire modal and masking filters are lower-order and wideband filters. As the number of channels increases, the value of ' l ' also increases, but the increment in ' l ' value is marginal as the filters are of lower-order. From the examples given in Section 5.5 and 5.6, it is noted that that ' n ' is 37.5% less than ' l ' and hence the proposed MSFRM based filter bank will result in the least complex channelizer when compared to all the above.

5.7 Summary

In this chapter, new reconfigurable low complexity multi-standard filters and filter banks based on the multi-band and masking technique as well as multi-stage frequency response masking (MSFRM) technique have been presented. The proposed architectures can be easily reconfigured for multi-mode operation. The multi-band filtering and masking technique allows complexity reduction when compared to filters realized using Parks McClellan method. The reconfigurability is achieved by swapping the masking filter coefficients according to current mode of operation using LUT as they are of lower-order nature. The proposed multi-band and masking approach offers an average gate count reduction of 26.5% over the PC approach for the design examples. The average reduction of power consumption achieved using the multi-band and masking approach over the PC approach is 12.2%. The second method proposed is a low complexity technique of MSFRM for the realization of reconfigurable channel filters. The complexity of MSFRM technique is reduced by eliminating the redundancy of the various stages of masking filters. The proposed MSFRM based filter bank architecture can extract non-uniform bandwidth and very narrow bandwidth channels compared to conventional filter banks. Design examples show that the proposed MSFRM based channel filter offers 47.9%

reduction in complexity compared to SSFRM based channel filter. Synthesis results show that the proposed MSFRM based channel filter offers average gate count reduction of 36.5% and power reduction of 18.1% with a slight increase in delay of 1.5% when compared to SSFRM based architecture. Unlike DFT filter banks, the proposed MSFRM based filter banks cannot fully shift the decimators prior to filtering. The decimators can only be shifted ahead of either modal or masking filter. Hence the proposed filter bank may need to work at higher sampling rate. This will lead to increased dynamic power consumption compared to that of DFT filter banks. But it may be noted that the area complexity of the proposed filter bank is 58.5% less when compared to DFT filter banks. Moreover the proposed multi-stage FRM filter bank is able to extract non-uniform channels whereas conventional DFT filter bank is incapable of performing non-uniform channel extraction. It has been reported that beyond 65 nm CMOS technology, the static power and leakage power is greater than dynamic power [84]. As the area complexity of the proposed filter bank is substantially less than that of DFT filter banks, the static and leakage power of the proposed filter bank will also be substantially less. Hence the total power consumption of the proposed filter bank (dynamic power + static power) would still be less than conventional DFT filter bank.

Even though the filters and filter banks based on the FRM technique are substantially less complex and easily reconfigurable, they are still unable to extract channel bandwidth which keeps a fractional factor with the modal filter bandwidth. In the next chapter, a new method, known as decimation interpolation and masking approach is proposed, to design filters, which can extract fractional bandwidth channels with low complexity and high level of reconfigurability.

Chapter 6

A Channel Filter Architecture for Fractional Bandwidth Channel Extraction using Decimation, Interpolation and Masking

In SDR channelizer, the channel filters may need to extract radio channels that can have bandwidths which are related by fractional factors with the modal filter bandwidth. The multi-band and masking and MSFRM method explained in Chapter 5, are not able to extract channels which have fractional relationship with modal filter bandwidth. In this chapter, the primary aim is to propose a method which can extract channels, whose bandwidths have fractional relationships with the modal filter bandwidth. Let BW be the bandwidth of the modal filter. Using conventional MSFRM approach, only channels that have bandwidths equal to $1/L \times BW$ can be extracted, where L is the interpolation value. On the contrary, in this Chapter, a channel filter architecture is proposed which can extract channels that have bandwidths equal to $D/L \times BW$, where D is the decimation value. The inclusion of the additional parameter D increases the flexibility of the proposed channel filter, which enables fractional bandwidth extraction. The fractional bandwidth extraction is made possible by using a two-stage approach, where the first stage of the proposed architecture employs a low pass linear phase FIR filter, called the *modal filter*, whose frequency band edges can be varied by changing the decimation and interpolation values. By varying decimation and interpolation values of the modal filter, bandwidths that are non-integer multiples of the modal filter bandwidth can be achieved. Thus the first stage will provide bandwidth adaptation, i.e., tuning of frequency bands corresponding to the channel spacing of a distinct communication standard. Based on the time-varying channels, the desired channel is adapted from the decimated and interpolated filter output using appropriate masking filters at the second stage. Thus the proposed channel filter architecture for a k -mode transceiver has a single fixed-coefficient modal filter at the front-end that can realize k distinct passbands (by varying the decimation and interpolation factors), where the width of each passband corresponds to the channel spacing of a distinct communication standard. At the back-end, the architecture has a

reconfigurable masking filter whose passband frequency edges are changed according to the current standard of operation.

6.1 Proposed Decimation, Interpolation and Masking (DIM) based Channel Filter

The proposed architecture is based on a combination of decimation, interpolation and frequency masking and hence named as DIM method. The procedure starts with the design of *modal filter* for the first stage of the two-stage architecture. The coefficients of modal filter are initially decimated by a factor of D which results in a frequency response whose BW and transition bandwidth (TBW) are D times larger than that of the original modal filter. The resulting filter is interpolated by a factor of L resulting in an $(L+1)$ -band filter frequency response, with BW and TBW $1/L$ times than that of the modal filter frequency response. The decimation-interpolation operation will alter the BW and TBW of each of the above obtained band by a factor of D/L compared to that of the modal filter frequency response. The desired channel can be extracted from these multiple bands by selectively masking the unwanted channels, using suitably designed masking filters. As the masking filters have large TBWs, their complexities are low. The principle of frequency response masking (FRM) was originally introduced in [63], and in Chapter 5, a multi-stage FRM based channel filter and filter bank is proposed which offers only one degree of freedom (factor, L) to change the TBWs and bandwidths of channels. But the work proposed in this chapter, DIM method, provides two degrees of freedom, D and L , which can be changed independently. Therefore this method significantly enhances the flexibility of the filter architecture to adapt with the channel spacing of more communication standards which are related to the modal filter bandwidth by a fractional factor.

6.1.1 Design Flow of the Proposed Channel Filter

The design algorithm of the proposed reconfigurable channel filter architecture is summarized below. All the frequency edges mentioned are frequencies normalized with respect to sampling frequency. Hence maximum value for a given frequency f_s is equal to $f_s/(2.f_s) = 0.5$, which is the normalized Nyquist frequency.

1. Generate an N -tap odd length modal filter, for a given passband and stopband specification. The passband and stopband frequency edges of the modal filter are

indicated as f_{ap} and f_{as} respectively. The modal filter order is taken as an odd value so that the complementary delay ($CD = (N_{modal} - 1) \times L/2$) value is an integer. In order to maintain the odd length of the modal filter after decimation by D , it must be ensured that modulo operation of the modal filter length (N) and D should be equal to 1 ($mod(\text{length of the modal filter } (N), D) = 1$). If the above condition is not satisfied, increase the value of N by 1, so as to satisfy the same.

2. The frequency response of the N -tap modal filter ($F_a(z)$) should be well within the normalized Nyquist frequency as shown in Fig. 6.1(a). This can be achieved by ensuring that the stopband edge of the modal filter, f_{as} , is within a factor of D of the normalized Nyquist frequency, i.e., $f_{as} \leq 0.5/D$ as given in [68]. The complementary response for the modal filter is shown using the dotted line in Fig. 6.1(a).

Figure 6.1 Frequency response illustration of the proposed DIM based reconfigurable channel filter.

3. Select suitable values for D , such that $D \leq 0.5/f_{as}$. The decimated modal filter response and its complementary response are shown in Fig. 6.1(b) as solid as dotted lines respectively. It can be noted that the BW and the TBW of the

decimated responses is D times larger than that of the modal filter frequency response. Select values of L , depending on the channel to be extracted. Fig. 6.1(c) shows the $(L+1)$ -band response. The decimation and interpolation operation of the modal filter response results in a multi-band response of BW, D/L times that of the original modal filter BW. The BW of each of the multi-band channels is identical to that of the channel spacing of the desired communication standard. It should be noted that there is an upper limit on the value of D as stated in step 1, beyond which aliasing can occur. The upper limit of D value should satisfy, $D.f_{as} \leq 0.5$.

4. Design an appropriate masking filter to extract the desired channel by masking unwanted channels from the multi-band response. Let f_p and f_s be the passband and stopband edges of the desired channel to be extracted, and f_{ap} and f_{as} be the passband and stopband edges of the modal filter. The passband and stopband frequency edges for the masking filter and the complementary masking filters are f_{map} , f_{mas} , f_{mcp} and f_{mcs} respectively. They are designed using the expressions given by (6.1):

$$\left. \begin{aligned} f_{map} &= f_p, \\ f_{mas} &= (\lfloor f_p * L \rfloor + 1 - (f_{as} * D)) / L, \\ f_{mcp} &= (\lfloor f_p * L \rfloor - f_{ap} * D) / L, \\ f_{mcs} &= f_s. \end{aligned} \right\} \quad (6.1)$$

The frequency response of the masking filters is shown in Fig. 6.1(d) and the over-all frequency response after DIM scheme is shown in Fig. 6.1(e).

Figure 6.2 The proposed DIM based reconfigurable channel filter for SDR.

The proposed architecture of the channel filter is shown in Fig. 6.2, where $F_a(z^{L/D})$ represents modal filter, $F_c(z^{L/D})$ represents complementary filter and $F_{MA}(z)$ and $F_{MC}(z)$ represent respective masking filters. The architecture can be explained using an example as follows. Let the specifications of the modal filter are passband edge, $f_{ap} = 0.1$ and stopband edge, $f_{as} = 0.125$. Following the design rules summarized in step 3 of Section 6.1.1, the maximum value of D is 4. The maximum value of L is fixed as 4 in this illustrative example. The length of the FIR filter can be obtained from the expression (6.2) [65]: where δ_p is the peak passband ripple, δ_s is the peak stopband ripple and $(f_s - f_p)$ is the TBW normalized to unity. Assume $\delta_p = 0.1$ dB and $\delta_s = -60$ dB for all the subfilters in Fig. 6.2.

$$N = \frac{-2 \times \log_{10} (10 \times \delta_p \times \delta_s)}{3(f_s - f_p)} - 1 \quad (6.2)$$

From (6.2), the length of the modal filter, N_{modal} , is obtained as 103. Taking into consideration the condition in the design rules for maintaining the odd filter length after decimation for the D values 1 to 4, the filter length is increased to 109. Two levels of reconfigurability, namely, filter level and architectural level can be achieved using the proposed architecture shown in Fig. 6.2. The filter level reconfigurability is achieved by reconfiguring the modal filter for different values of D and L , in order to realize narrow BW channels. Mux-3, shown in Fig. 6.2, is utilized along with the complementary response to exploit the architectural level reconfigurability to realize larger BW channels.

6.1.2 Filter Level Reconfigurability

Filter level reconfigurability can be achieved by dynamically changing the interpolation and decimation factors, L and D respectively, of the modal filter, to extract channels of different bandwidths. Here only the narrow BW output channels are utilized, obtained after decimation and interpolation of modal filter ($F_a(z^{L/D})$) followed by subsequent masking filter ($F_{MA}(z)$), without taking the complementary section of the proposed architecture. The modal filter architecture exploiting filter level reconfigurability is shown in Fig. 6.3. The modal filter is an FIR filter and the filter coefficients are shown as $h_0, h_1, \dots, h_{N-1}$. The symmetrical FIR filter structure along with the architecture to change the decimation and interpolation values of the modal filter is shown in Fig. 6.3.

Figure 6.3 Architecture of modal filter, $F_a(z^{LD})$. The filter coefficients are indicated as h_0 , h_1 etc and the filter is a symmetric FIR filter.

The decimation factor D can be varied using the multiplexers indicated as Mux-1. For example, when $D = 2$, every alternate coefficients are omitted using Mux-1. The select value for the Mux-1 is stored in a look-up table (LUT). Let the select value be '0', Mux-1 passes the $In-2$ input and for select value '1', Mux-1 passes the $In-1$ input. Then for $D = 1$, the first stored value in the LUT, "0 0 0 0 0 ... 0", is sent to Mux-1. The word length of the stored value in LUT is equal to $(N - 1)$, where N is the filter length. For $D = 2, 3$, etc, the LUT output will be "1 0 1 0 1 01 0", "1 1 0 1 1 01 1 0" etc. Note that decimation by D reduces the magnitude of the passband frequency response to a value less than 0 dB. In order to overcome this passband attenuation, the output is multiplied by D to obtain Y , so that the passband gain is restored to 0 dB. The multiplication by D operation would also reduce the stopband attenuation, which could deteriorate the out of band rejection of the filter. Therefore, based on the desired stopband attenuation of the final filter, the original modal filter should be designed with larger stopband attenuation keeping in account of the deterioration of the final filter's stopband response. That is, the modal filter decimated by a factor D , should have a stopband attenuation, $\delta_{s(modal)} = \delta_{s(final\ filter)} / D$. The value of L can be altered from 1 to 4 using the multiplexer, Mux-2, by changing the select signal from "00" to "11", i.e. for L

= 3 the select signal for Mux-2 will be “10”. The output Y_1 of the modal filter can be extracted through Mux-3 in Fig. 6.3, where the select signal value is “1”.

For a generalized scheme of the proposed architecture, $D.L$ number of channels, each of distinct bandwidth, can be obtained. As the maximum value of D and L is set as 4 in this example, $D.L = 16$ channels are obtained at the output Y_1 , each of distinct bandwidth, by utilizing the filter level reconfigurability alone. The channel of desired bandwidth can be obtained from above multi-band output by masking unwanted channels using masking filters which is designed with large TBWs. The lengths of the masking filters are shorter on account of their large TBWs and consequently, they are of low complexity compared to modal filters. The specifications of all the 16 channels along with their masking filter specifications are given in Table 6.1. Only 3 masking filters are required to extract the 16 channels listed in Table 6.1.

Table 6.1
Specifications of the channels for the example using filter level reconfigurability.

No. of channels	L	D	f_p	f_s	$[f_{map}, f_{mas}]$ of Masking filter
1	1	1	0.1	0.125	[0.1,0.5]
2	1	2	0.2	0.25	[0.2,0.5]
3	1	3	0.3	0.375	[0.3,0.5]
4	1	4	0.4	0.5	[0.4,0.5]
5	2	1	0.05	0.0625	[0.05, 0.4375]
6	2	2	0.1	0.125	[0.1,0.5]
7	2	3	0.15	0.1875	[0.15,0.3125]
8	2	4	0.2	0.25	[0.2,0.5]
9	3	1	0.033	0.04166	[0.033,0.2917]
10	3	2	0.066	0.0833	[0.066,0.25]
11	3	3	0.1	0.125	[0.1,0.5]
12	3	4	0.133	0.166	[0.133 0.166]
13	4	1	0.025	0.03125	[0.025,0.2188]
14	4	2	0.05	0.0625	[0.05, 0.4375]
15	4	3	0.075	0.09375	[0.075,0.3125]
16	4	4	0.1	0.125	[0.1,0.5]

The specifications of the 3 masking filters are:

1. For channels 1, 2, 3, 4, 6, 8, 11 and 16, only one masking filter with specifications, $f_{map} = 0.4$ and $f_{mas} = 0.5$ and masking filter length using (6.1), $N_{mask1} = 25$ is needed.

2. For channels 5, 7, 9, 10, 13, 14 and 15, only one masking filter with specifications, $f_{map} = 0.15$ and $f_{mas} = 0.2188$ and $N_{mask2} = 37$ is needed.
3. For channel 12, another masking filter which has the same specification as that of the desired channel and $N_{mask3} = 79$ is needed.

Hence with a total filter length of $N_{modal} (=109) + N_{mask1} + N_{mask2} + N_{mask3} = 250$, 16 channels can be extracted, each of distinct bandwidth, by exploiting filter level reconfigurability of proposed scheme.

6.1.3 Architecture Level Reconfigurability

Architecture level reconfigurability can be exploited with the help of Mux-3 as shown in Fig. 6.2. When the select signal for Mux-3 is '0', the complementary response obtained at the output of $F_c(z^{L/D})$ is suitably masked and added with the output Y_1 to obtain another set of distinct wide bandwidth channels at Y_2 . The TBW and BW of the $(L+1)$ band response at the modal filter output and the complementary filter output are altered by a factor of D/L (where D and L can be varied independently), thus allowing more flexibility over reconfigurable FRM. The complementary delays can be obtained using the equation $\frac{L}{2} \left(\frac{N_{modal} - 1}{D} \right)$, where N_{modal} is the length of the modal filter. This implies that, $(N_{modal} - 1)$ should be completely divisible by D and this criterion is taken into account while designing the modal filter in the design rules of the proposed architecture. Using architectural reconfigurability, $(L + 2) = 6$ channels can be extracted, each of distinct bandwidth at output Y_2 , as shown in Table 6.2.

The lengths of the masking filters are very short as their passband and stopband frequency edges are sufficiently wide apart as indicated in Table 6.2. The channels 1, 2, and 3 can be extracted using the masking filter $F_{MA}(z)$ with frequency edges [0.4333, 0.5] and $F_{MC}(z)$ with frequency edges [0.3, 0.375]. The channels 4, 5, and 6 can be extracted using the masking filter $F_{MA}(z)$ with frequency edges [0.325, 0.4062] and $F_{MC}(z)$ with frequency edges [0.225, 0.2813]. Hence the total length of the masking filters for extracting the channels given in Table 6.2 is $37 \times 2 + 46 \times 2 = 166$. The masking filters can be reconfigured according to the required output response by swapping the filter coefficients stored in the LUT. The size of the LUT is small as the large

transition-band width masking filters are of significantly lower order compared to modal filters.

Table 6.2

Specifications of the channels for the example using architecture level reconfigurability.

No. of channels	L	D	f_p	f_s	$[f_{map}, f_{mas}]$ of masking filter	$[f_{mcp}, f_{mcs}]$ of complementary masking filter
1	3	1	0.3667	0.3750	[0.3667,0.5]	[0.3,0.375]
2	3	2	0.4	0.4167	[0.4,0.5]	[0.266,0.4167]
3	3	3	0.4333	0.4583	[0.4333,0.5]	[0.2333,0.4583]
4	4	1	0.2750	0.2813	[0.275,0.4688]	[0.225,0.2813]
5	4	2	0.3	0.3125	[0.3,0.4375]	[0.2,0.3125]
6	4	3	0.325	0.3438	[0.325,0.4062]	[0.175,0.3438]

In conclusion, for the above example with maximum value of D and L as 4, $D.L = 16$ channels can be extracted using filter level reconfigurability and $L+2 = 6$ channels using architecture level reconfigurability. Thus a total of $(D.L + (L+2)) = 22$ channels, each of distinct bandwidths, can be extracted. If any other channel bandwidth specification is required, there are two possibilities. (1) Choose values of L larger than 4 (here for simplicity L values up to 4 is only shown) so as to obtain the desired channel bandwidths using filter level or architectural level reconfigurability. (2) Reconfigure the modal filter using LUT coefficient swapping method explained earlier in case of masking filter as the complexity of modal filter is lower than that of the channel filter realized using Parks McClellan technique.

6.2 Implementation Results

The proposed DIM based channel filter architecture is implemented on Xilinx Virtex 2v3000ff1152-4 FPGA associated with the dual DSP-FPGA Signalmaster kit provided by Lyrtech [66]. The proposed channel filter is tested using simulated Gaussian noise inputs. The objective is to test the functionality of the architecture by selecting the desired channel to pass through the masking filter. The design diagram for proposed channel filter architecture taken directly from the Simulink is shown in Fig. 6.4. The bit-stream of the proposed simulation architecture was generated using the Xilinx system generator. The generated bit stream can be downloaded to FPGA using board configuration block and it appears as a ‘Xilinx hwcosim block’. The performances of

the bit stream and the simulation architecture were checked to ensure that they are identical. The power consumption figures were calculated using X-power.

Figure 6.4 Simulink diagram of the proposed DIM based channel filter architecture.

Table 6.3
Implementation results for a single channel extraction (Channel-4).

Standard	Total equivalent gate count	Total power (mW)	Total delay (ns)
Proposed DIM architecture	476,636	396.5	17.035
PC architecture	533,451	421.3	16.71

Table 6.3 shows the implementation results. The proposed DIM method is compared with per-channel (PC) method for generating channel 4 obtained using architectural level reconfigurability listed in Table 6.2, as the extraction of channel 4 needs a filter with the most stringent TBW and consequently highest filter length. The filter length for extracting channel 4 by PC method is 416. The total filter length for the proposed architecture is 109 (modal filter length) + 2x46 (masking filter length for channel-4) = 201, which is 51.68% less complex than the PC approach. Section 6.1.2 describes the

LUT based method for reconfiguring the shorter length masking filters. For the current example of channel 4, masking filter length is higher compared to other channels. Thus it can be considered that the worst case filter length is taken for the implementation of the proposed architecture. The proposed DIM based channel filter offers an average gate count reduction of 10.65% over the PC approach for the channel 4 given in Table 6.3. The average reduction of power consumption achieved using the proposed DIM based channel filter over the PC approach is 5.88%. Delay measurement given in Table 6.3 is the net combinatorial time for the first sample to appear at the output since the first sample is fed to the architecture. From Table 6.3, it can be noted that, the proposed architecture has a small increase in delay (1.9%) compared to the PC architecture. However, this increase in delay is very small compared to the area and power savings mentioned above. Moreover if fixed coefficient filters are considered, it must be noted that only one channel (channel 4 in this case) can be extracted using the 416-tap filter in PC architecture, the proposed architecture can extract all the 22 channels with a total filter length of 416 ($N_{modal} + N_{mask1} + N_{mask2} + N_{mask3} = 250$ in Table 6.1 + Total N_{mask} of 166 in Table 6.2 = 416). Even though, only one channel needs to be extracted at a given time in an SDR handset, once a filter circuit is designed, that can extract 22 distinct channels with a total filter length of 416 using the proposed method method, the same architecture can be used to extract any of the 22 channels using filter level and architecture level reconfigurability. On the contrary, conventional PC architecture would require 22 separate fixed coefficient filters to accomplish this task. Consequently, the total area of the PC-architecture-based channel filter would increase substantially and the total gate count and power consumption figures in Table 6.3 will be several orders higher than that of the proposed architecture.

An efficient alternative to PC architecture, as a better and fairer method for comparing the proposed DIM architecture, is a reconfigurable FIR filter employing the reconfigurable architectures in [55-57]. Such a reconfigurable filter will have a total filter length of 416 corresponding to the specification of the most stringent band to be extracted (which is same as the most stringent specification filter in PC architecture). Thus using a 416-tap reconfigurable FIR filter and extra reconfiguration hardware, any of the 22 channels can be extracted. However, the area-time complexity of reconfiguration hardware in such reconfigurable filter implementations [55-57] is substantially larger than that of the proposed method. A comparison of the proposed

DIM based channel filter architecture is done with the reconfigurable multiplier block (ReMB) filter architecture in [57]. The ReMB generates all the coefficient products and a multiplexer will select the desired products depending on the input. The area of the ReMB block of an N -tap filter is proportional to the multiple constant multiplication (MCM) block of a transposed form N -tap FIR filter with an extra overhead of a N input multiplexer. But the length of the filters used in the proposed method is substantially less than that in [57]. The area and delay measurement for the 32-tap half band filter design example in [57] shows that the delay is 23.6 ns and number of LUTs utilized is 190, on a Virtex XCV300BG432-4 FPGA. The delay in [57] is directly proportional to the length of the filter as ReMB follows a serial implementation to reduce the area count. If ReMB architecture in [57] is employed for extracting any of the 22 channels, then a 416-tap reconfigurable filter needs to be implemented, which is almost 26 times (13×2 as the example filter in [57] is a half band filter) larger than the original filter length. Then, using ReMB, the area count will be approximately 4940 LUT and delay will be approximately 613.6 ns, whereas the proposed channel filter gives an area count of 7653 LUT and delay of 17.035 ns only. The reconfiguration cost of the reconfigurable FIR filter in [57] is the cost of generating all the possible product combinations of the input and the coefficients and the reconfiguration time is proportional to the order of the filter (here $N = 416$) whereas, in the proposed method only the select signal to the multiplexers are changed. Hence the proposed method offers a substantially smaller reconfiguration delay compared to [57]. The reduction of area-time complexity of the proposed DIM based channel filter over [57] is 95.7%.

6.3 Design Example

In this section, a design example of the proposed reconfigurable architecture for dual-standard Code Division Multiple Access (CDMA)/Wideband CDMA (WCDMA) channels for an SDR handset is presented. The sampling rate chosen is 150 MHz. The channel filters can extract 1250 kHz CDMA channels and 5 MHz WCDMA channels from the input signal after down sampling by a factor of 4. The passband and stopband edges are 1250 kHz and 1300 kHz respectively for CDMA and 5000 kHz and 5050 kHz for WCDMA. The peak passband ripple is chosen as 0.1 dB. The filter stopband specification is chosen as -48 dB for both the standards. Reconfigurability can be achieved as follows. The channel specifications for CDMA is, $f_p = (1250 \times 4/150000)$

$= 0.033$ and $f_s = (1300 \times 4/150000) = 0.035$ at the output where 150 kHz is the sampling frequency and the down sampling factor is 4. Now considering WCDMA, a channel with specifications $f_p = (5000 \times 4/150000) = 0.133$ and $f_s = (5050 \times 4/150000) = 0.135$ is needed at the output. The CDMA/WCDMA reconfigurable channel filter can be designed as in the illustrative example in Section 6.1, employing different values for L and D . For CDMA, the values of L and D corresponding to channel 9 are shown in Table 6.1. Hence the select values for Mux-2 in Fig. 3 is “10” ($L = 3$) and for Mux-1, the LUT output is all zeros ($D = 1$). For WCDMA, the values of L and D corresponding to channel 7 are shown in Table 6.1. Hence the select value for Mux-2 is “01” ($L = 2$) and for Mux-1, the LUT output is “1 1 0 1 1 01 1 0” ($D = 3$). The CDMA and WCDMA signals can be extracted using the channels as shown in Figures 6.5 and 6.6 respectively. The output Y of the modal filter ($F_a(z)$ in Fig. 6.2) after the desired values of D and L is shown by the bold line in Figures 6.5(a) and 6.6(a). The masking filter characteristics for selecting the required channels for CDMA and WCDMA are indicated by the dotted lines in Figures 6.5(a) and 6.6(a). The extracted CDMA and WCDMA are shown in Figures 6.5(b) and 6.6(b) respectively. For both CDMA and WCDMA, the output is taken at Y_1 (after $F_{MA}(z)$ in Fig. 6.2) and hence the select signal for Mux-3 is 1. The LUTs of masking filters are loaded with coefficients corresponding to $[f_{mp} = 0.033, f_{ms} = 0.2917]$ for CDMA and $[f_{mp} = 0.15, f_{ms} = 0.3125]$ for WCDMA respectively as shown in Table 6.1.

Figure 6.5(a) Magnitude response of the modal filter and masking filter for extracting CDMA signal.

Figure 6.5(b) Magnitude response of the channel filter for extracting CDMA Signal.

Figure 6.6(a) Magnitude response of the modal filter and masking filter for extracting WCDMA Signal.

If PC approach was employed to adapt to the channel of interest, a channel filter of length of 1314 would be needed for extracting either CDMA or WCDMA channel. In the proposed method, the total length for CDMA/WCDMA adaptation only requires $109 (= N_{modal}) + 9 (= N_{mask} \text{ for CDMA}) + 15 (= N_{mask} \text{ for WCDMA}) = 133$. Hence the proposed method offers about 90% reduction in complexity compared to that of PC approach. If the reconfigurable FIR filter approach in [57] is used, then a ReMB block is needed which realizes all the combinations of input with the 1314 filter coefficients.

The delay will be approximately 41 times larger than the proposed method, using a linear mapping of the 32-tap half-band filter example in [57]. In the proposed method, if none of the lowpass and bandpass outputs indicated in the above example satisfy the requirements of a new communication standard, the modal filter specifications needs to be changed.

Figure 6.6(b) Magnitude response of the channel filter for extracting WCDMA Signal.

6.4 Functionality Test

The functionality of the proposed DIM based channel filter architecture for the CDMA/WCDMA design example is tested as follows. The specifications are same as that of the design example presented in Section 6.3. The sampling rate chosen is 20 MHz. Gaussian white noise of sampling frequency 20 MHz is given as the input of the proposed channel filter architecture for simulation. Fig. 6.7(a) shows the frequency spectrum of the input signal and the CDMA spectrum extracted using the proposed DIM based channel filter. Fig. 6.7(b) shows the extracted WCDMA spectrum along with the input spectrum. The spectrum of CDMA satisfies the design specifications ($f_p=1250$ kHz and $f_s=1300$ kHz $\delta_p = 0.1$ dB and $\delta_s = -48$ dB) and WCDMA satisfies the design specifications ($f_p = 5000$ kHz, $f_s = 5500$ kHz and $\delta_p = 0.1$ dB and $\delta_s = -48$ dB) as shown in Figures 6.7(a) and 6.7(b). The average passband error for CDMA and WCDMA is 0.032 dB and 0.067 dB respectively. Thus the proposed reconfigurable

DIM based channel filter is capable of efficiently adapting to a desired channel from a set of channels of various communication standards.

Figure 6.7(a) Input signal spectrum and CDMA spectrum extracted using DIM based channel filter.

Figure 6.7(b) Input signal spectrum and WCDMA spectrum extracted using DIM based channel filter.

6.5 Summary

In this chapter, new reconfigurable low complexity multi-mode filters based on the Decimation Interpolation and Masking (DIM) technique have been presented. The proposed architectures are inherently less complex and offer two levels of reconfigurability: 1) at architectural level and 2) at filter level. The DIM based channel filter can extract channels, whose bandwidths are related by fractional factors. The proposed architecture is able to realize $D.L + (L+2)$ channels using the same modal filter, where D and L are the decimation and interpolation factors respectively. Complexity analysis with respect to the design example shown clearly indicates that the proposed architecture offers average filter length reduction of 90% compared to that of PC method. The implementation results show that method offers an average gate count reduction of 10.7% over the PC approach for the channel-4 in the design example. The average reduction of power consumption achieved using the proposed method over the PC approach is 5.9%. The proposed DIM based channel filter architecture offers 95.7% area-time complexity reduction when compared to the reconfigurable multiplier block filters [57]. In this proposed technique, the coefficients of the masking filter have to be reconfigured depending upon the channel-of-interest. If the channel-of interest is difficult to obtain by varying D and L values for a given modal filter, then in that case the modal filter also should be reconfigured.

In this thesis, three methods are proposed for channel filtering: - (1) Multi-band filtering and masking based channel filter, (2) Multi-stage frequency response masking (MSFRM) based channel filter and (3) Decimation-interpolation and masking (DIM) based channel filter. Following explanation distinguishes clearly the scenarios where the three proposed methods can be used. Out of the three methods multi-band filtering and masking method will have the highest complexity, as the method makes use of a multi-band filter realized for the most stringent specification. Multi-band filtering and masking method is useful when the channels to be extracted are scarcely located in a wideband frequency spectrum. The advantage of multi-band filtering and masking method is that, absolute control over the passband width and location of centre frequency is possible, with prior knowledge of the wideband spectrum. MSFRM based channel filtering gives the least complexity solution for channels to be extracted whose

bandwidth is $1/L$ (L = interpolation value) of the modal filter bandwidth. DIM based channel filtering extracts channels whose bandwidth shares a fractional relation (D/L , D = decimation value and L = interpolation value) with the modal filter bandwidth. This extra reconfigurability property comes on the cost of higher complexity for the modal filter when compared to MSFRM based approach. The next chapter in this thesis is based on a high speed and novel method for reconfiguring the masking filters based on Residue Number system (RNS).

Chapter 7

A Reconfigurable High-Speed Residue Number System Based FIR Filter

In Chapters 5 and 6, methods for reconfigurable channel filtering by new low complexity techniques using lower-order subfilters were presented. The masking subfilters of the proposed channel filters in Chapter 5 and 6, need to be reconfigured according to the bandwidths of the channels-of-interest. In this chapter a novel technique of high-speed and reconfigurable FIR filter design using Residue number systems (RNS) is proposed. These high-speed reconfigurable filters can be used as the masking filters for the channel filters proposed in Chapter 5 and 6. They can also be used as general purpose reconfigurable FIR filters serving as channel filters. The proposed design is essentially high-speed in nature due to the inherent parallelism property of RNS arithmetic. This enables the channel filters which are placed immediately after the ADC to operate at the highest speed in the digital front-end. All reconfigurable RNS FIR filters proposed in literature focus on reconfigurability for different moduli set for the same set of coefficients (consequently the same frequency response) with the objective of achieving different dynamic ranges. The novelty of the RNS FIR filter design proposed in this chapter is that, for a given moduli set, the filter architecture can be reconfigured to satisfy the frequency response specifications of channel filters of different wireless communication standards.

7.1 A Review on RNS - FIR Filters

Filter realizations using alternative number systems such as Residue number system (RNS) have been investigated in literature. RNS is suitable for implementation of high-speed digital signal processing due to their inherent parallelism, modularity, fault toleration and local carry propagation properties [69]. Arithmetic operations like multiplication and addition can be carried out more efficiently as RNS ensures localized carry propagation properties. RNS is particularly suitable for implementing FIR filters as the core operations in FIR filtering are multiplications and additions.

RNS is defined by a moduli set, which consists of p pair wise relatively prime integers $\{m_0, m_1, \dots, m_{p-1}\}$. The useful computational range M of such a number system, which is called the legitimate range, is defined by the product of all moduli in the moduli set, i.e. $M = \prod_{i=0}^{p-1} m_i$. A residue number system with valid range M is able to uniquely represent unsigned numbers in the range of $[0, M - 1]$, or signed numbers in the range of $[-(M-1)/2, ((M+1)/2)-1]$ for odd M , and $[-M/2, (M/2)-1]$ if M is even. These ranges are known as the dynamic ranges of RNS [69]. Any integer x within the dynamic range has a unique RNS representation that can be characterized by the list of its residues with respect to the moduli defined in the moduli set.

$x \xrightarrow{RNS} (\langle x \rangle_{m_1}, \langle x \rangle_{m_2}, \dots, \langle x \rangle_{m_{p-1}})$ where $\langle x \rangle_{m_i}$, denotes the operation [69]:

$$\langle x \rangle_{m_i} = \begin{cases} x \bmod m_i & \text{for } x \geq 0 \\ (M - x) \bmod m_i & \text{for } x < 0 \end{cases} \quad (7.1)$$

Addition, subtraction and multiplication can be performed in RNS domain on the residue representation of the operands. All the above operations, on different m_i (moduli) are done in parallel as shown below.

$$z = x \circ y \rightarrow \begin{cases} z_{m_1} = \langle x_{m_1} \circ y_{m_1} \rangle_{m_1} \\ z_{m_2} = \langle x_{m_2} \circ y_{m_2} \rangle_{m_2} \\ \dots \\ z_{m_{p-1}} = \langle x_{m_{p-1}} \circ y_{m_{p-1}} \rangle_{m_{p-1}} \end{cases} \quad (7.2)$$

where the operator ' $\circ$ ' can be addition, subtraction or multiplication. Consequently, operations on large wordlengths can be split into several modular operations executed in parallel with shorter wordlengths. Since two's complement is the most common number representation, conversions between residue and binary representations are required. Mapping from the RNS back to the binary representation is defined by the Chinese Remainder Theorem (CRT) [69].

A Finite Impulse Response (FIR) filter of order N is described by the expression

$$y(n) = \sum_{k=0}^{N-1} H_k x(n-k) \quad (7.3)$$

The implementation of RNS based FIR filter can be shown by combining equations (7.2) and (7.3) to form (7.4) [69].

$$\begin{cases} y_{m_1}(n) = \left\langle \sum_{k=1}^N \left\langle H_{m_1}(k) \cdot x_{m_1}(n-k) \right\rangle_{m_1} \right\rangle_{m_1} \\ y_{m_2}(n) = \left\langle \sum_{k=1}^N \left\langle H_{m_2}(k) \cdot x_{m_2}(n-k) \right\rangle_{m_2} \right\rangle_{m_2} \\ \dots\dots\dots \\ y_{m_{p-1}}(n) = \left\langle \sum_{k=1}^N \left\langle H_{m_{p-1}}(k) \cdot x_{m_{p-1}}(n-k) \right\rangle_{m_{p-1}} \right\rangle_{m_{p-1}} \end{cases} \quad (7.4)$$

FIR filtering in RNS domain is done using multiple modulo m_i FIR filter blocks working in parallel, as shown in Fig. 7.1. Each of them is implemented with additions and multiplications *mod* m_i as shown in Fig. 7.2 where ‘ D ’ represents inter-tap delay. A drawback of RNS is the need for converters to/from the conventional number system (two’s complement) termed forward/reverse converters respectively. However, this conversion overhead has only a small impact for high-order filters such as the ones needed for channel filtering in SDR receivers [74], because the conversion hardware is only a small fraction of the filter hardware in such cases. Section 7.3 shows that the proposed architecture is not only good for higher-order channel filters and but also for lower-order masking filters.

Figure 7.1. FIR filter implementation using RNS.

Figure 7.2. $|FIR|_{mi}$ Structure.

A fast RNS based communication receiver design and implementation using Field programmable gate arrays (FPGAs) is presented in [70]. RNS arithmetic operations are implemented in [70] as a set of concurrent primitive operations with non-communicating small wordlength channels. A hybrid RNS adaptive filter was introduced in [71] where the adaptive filter implementation is done partly in RNS and partly in two's complement. This enables the architecture to have the advantage of area and speed of the hybrid RNS adaptive filter with respect to the two's complement one. Both [70, 71] discussed dedicated RNS filter architectures while [71] deals with the hybrid architecture using RNS. Reconfigurable computing in the RNS domain is a rather new field. Most research in this field has been focused on Variable word length (VWL) RNS processors [72], which provide reconfigurability in terms of dynamic range by providing the ability to turn off the channels that are not required. This kind of reconfigurability can offer variable precision realization of FIR filters. A multi-modulus modulo multiplier based on sharing of common resources for the three moduli set $\{2^n-1, 2^n, 2^n+1\}$ has been investigated in [73]. Although reconfigurability has been addressed in RNS literature from the perspective of realizing filters with variable coefficient wordlengths and multi-modulus modulo multipliers, hardly any RNS based work addressed the problem of designing filter architectures for different frequency response specifications as required in the case of multi-standard wireless communication receivers. In such receivers, very-high-order FIR channel filters are required to meet the stringent adjacent channel attenuation specifications of wireless communication standards. It has been shown in [74] that such higher-order FIR filters can be implemented with low hardware complexity in RNS domain. However, reconfigurability of filters for multi-standard operation was not addressed in [74]. In

this chapter, an RNS-FIR filter is proposed whose architecture can be reconfigured to satisfy the filter frequency response specifications.

7.2 Proposed Reconfigurable RNS-FIR Filter

In this section, a reconfigurable filter architecture using RNS is presented. Here the reconfigurability is achieved by using the same architecture for multi-standard channel filters where the filter specifications and consequently the coefficients are distinct for each standard.

7.2.1 Proposed Reconfigurable Channel Filter Architecture

The coefficient values of FIR filters, catering for multiple standards are generally stored in look up tables (LUT). In conventional LUT based filter implementation [57], the filter coefficients are stored as such with its full precision. The multiplication is done as a one-to-one mapping of $(k_1 \times k_2)$ bits, where k_1 and k_2 are the possible combinations of input and coefficients respectively. This approach will consume huge search space in LUT and thus the area also increases. In the proposed method the coefficient values of filters for multiple standards are stored in a LUT with respect to their corresponding residue value. Hence the coefficient value to be stored in the LUT will enjoy less area as the wordlength is limited by the modulo value. The relevant coefficient set corresponding to each standard is fed to the architecture corresponding to the current mode of operation. The proposed reconfigurable filter is realized using the following example. The moduli values are fixed for a particular format, $\{2^n-1, 2^n, 2^n+1\}$ as $\{31, 32, 33\}$, for illustration purposes. The precision or wordlength of the filter is defined as the number of bits used to represent the filter coefficients. The wordlength is fixed for fixed moduli. The proposed RNS-FIR method aims to reconfigure the filter coefficients on the fly, corresponding to multiple standards, for a given word length.

It can be noted that for moduli based number systems such as RNS, the maximum possible value of the residue of any number in the modulus $\{m_1, m_2, m_3\} = \{31, 32, 33\}$ is $(m_3-1)=32$. Thus the residue result of any computation in the modulus m_1 can only have values from 0 to m_1-1 . This reduced range of residue values is exploited in the proposed method for hardware optimization when compared to the original range as taken by other LUT based algorithms [57]. Let the maximum modulus in a given

moduli set is m_{max} . The quantized input and the filter coefficient are converted to the corresponding moduli and let $m_{max} - 1$ be the maximum value for both, for the given moduli. Then, if all the bit combinations of $(m_{max} - 1) \times (m_{max} - 1)$ are computed, the net result can only have a maximum value of $(m_{max} - 1)$. Hence the output value of the modulo multiplier can vary only between 0 to $(m_{max} - 1)$. The interesting property is that for a given moduli, the input as well as any set of filter coefficient can vary only between 0 to $(m_{max} - 1)$, and hence the modulo multiplier needs to be implemented only once and it can be reused for any set of input and coefficients. Using this property the multiplier can be redesigned as a product encoder which selects the pre-computed multiplied result as output corresponding to the filter input and filter coefficient. The search space of the encoder is limited as the output value is again limited by the modulo value. There are redundant cases of the multiplications between filter input and coefficient when they share a commutative property (filter input $\times$ filter coefficient = filter coefficient $\times$ input.) The cases of a commutative multiplication property between input and coefficients are identified and eliminated in the proposed approach, to improve the area efficiency of the product encoder.

In addition to reducing the area of the filter circuit, a technique to integrate reconfigurability in FIR filter architecture is also proposed. The proposed scheme shown in Fig. 7.3 can be explained as follows. The precision or wordlength of input and filter coefficient for the moduli set of $\{31, 32, 33\}$ is 8. If the wordlength of the quantized input signal is less than 8 bits then a forward converter to convert the binary input number to RNS is not required. If quantized input has more than 8 bits, it is converted to respective residue value using a forward converter. The residue values for each coefficient set corresponding to a distinct communication standard can be calculated offline and stored in LUT and hence a forward converter is not required for filter coefficients. Thus, for a k -standard wireless communication receiver, the LUT will have modulo values of all the coefficients of k filters. The product encoder for the particular moduli selects the pre-computed product of the input signal with the corresponding filter coefficient stored in LUT. The results are added using modulo add operation for the corresponding moduli as shown in Fig. 7.3. The output of each FIR filters corresponding to each moduli, $|y(n)|_{m_i}$, is reverse converted to binary to obtain the filter output, $y(n)$. Note that forward and reverse converters are not shown in Fig. 7.3.

The moduli set is made fixed for a multi-standard receiver taking into account of the maximum dynamic range corresponding to the wireless standard that has the most stringent specification among the set of wireless standards. There is hardly any need of changing the dynamic range for the wireless standards for which the receiver is designed as the product encoder is designed for the most stringent dynamic range. When the mode of communication switches to a new standard, the coefficient set corresponding to the filter for new standard will be loaded in the LUT. In this case also, the same pre-designed encoder performs product selection. The modulo adder and the reverse converter are also designed corresponding to the pre-fixed moduli set. Thus the proposed high-speed architecture can work as a reconfigurable FIR filter, for a fixed moduli value.

Figure 7.3. Reconfigurable RNS-FIR filter for $mod\ m_i$.

The proposed RNS-FIR filter differs from the conventional RNS based FIR implementation in the following aspects. The proposed architecture is a reconfigurable architecture which can switch to a new communication standard with a fixed dynamic range. Existing RNS based FIR filter implementations are on variable dynamic range, and not for reconfiguration according to variable frequency specification, as needed by multi-standard communication receivers. The unique part of the proposed work is the design of product encoder for a given wordlength. This reduces the area-time complexity of the product encoder. On the contrary, conventional RNS methods design the multiplier for a generalized format for variable wordlength.

7.2.2 Product Encoder Synthesis Results

The product encoder has been implemented in TSMC 0.18 μ m CMOS standard-cell technology as a proof-of-concept. The architectures were coded in Verilog and functionally is simulated using MODELSIM SE. Synopsys Design Compiler (v2001.08) was used for the synthesis. The synthesis results and their comparison with RNS multiplier architectures available in literature [73], [75-79] are shown in Table 7.1. The area-time metrics of proposed $2^n + 1$ encoder for different bit-width ($n = 4, 8, 16$) and that of the multipliers in [73], [75-79] are summarized in Table 7.1.

Table 7.1
Synthesis result of product encoder.

(4bit x 4 bit)	Area (μm^2)	Time Delay (in ns)	Area-Time
Proposed $2^n + 1$ encoder	1350.5	1.27	1715.1
Efstathiou. et. al [76]	3724	1.34	4990.2
Vergos. et. al [77]	3206	1.09	3494.5
Wang et. al [78]	3939	1.51	5947.9
Ma et. al [79]	3134	1.64	5139.8
(8bit x 8 bit)	Area (μm^2)	Time Delay (in ns)	Area-Time
Proposed $2^n + 1$ encoder	9225	2.07	19095.8
Zimmermann. [75]	20232	2.47	49973
Efstathiou. et. al [76]	9685	1.98	19176.3
Vergos. et. al [77]	8819	1.96	17285.2
Shibu. et. al [73]	21231.75	3.05	64756.8
Wang et. al [78]	10427	2.19	22835.1
Ma et. al [79]	8830	2.32	20485.6
(16bit x 16 bit)	Area (μm^2)	Time Delay (in ns)	Area-Time
Proposed $2^n + 1$ encoder	32485	2.82	91607.7
Zimmermann. [75]	66580	3.68	245014.4
Efstathiou. et. al [76]	36541	2.65	96833.7
Vergos. et. al [77]	34252	2.61	89397.72
Shibu. et. al [73]	67236	3.69	248100.8
Wang et. al [78]	38619	2.94	113539.9
Ma et. al [79]	29856	3.09	92255

For 4 bit x 4 bit multiplication, both area and time metrics of the proposed product encoder outperforms the multiplier architectures in [76, 78, 79]. The proposed architecture is having a greater delay when compared to [77], but in terms of area x time (AT) complexity the proposed method outperforms [77] by 50 %. For 8 bit x 8 bit multiplication, the method offers lower delay and area when compared to the multipliers in [73, 75, 78, and 79]. The proposed reconfigurable product encoder is having a higher delay when compared to [76, 77], but taking the AT complexity, the architecture outperforms [76]. The AT complexity is least for the proposed method in 16 bit x 16 bit multiplication. On an average, the AT complexity results shows that, the proposed 2^n+1 product encoder architecture has 67% improvement over [73], 62.2% over [75] 23.43% improvement over [76], 35.8% improvement over [78] and 24.4% improvement over [79]. The proposed architecture results in slightly inferior AT complexity result, when compared to [77], for the 8 bit and 16 bit multiplication. This is basically because the method in [77] is a dedicated multiplier optimized for area and delay. In the proposed architecture, AT reductions are made possible due to the elimination of redundant multiplications and reduction in the output search space of the product encoder by fixing the moduli to the most stringent dynamic range.

7.3 Design Example and Synthesis Results

In this section, the synthesis results of the FIR filters are presented. Both higher-order channel filter and lower-order filters like the masking filters in Chapter 5 and 6 are presented. At first an FIR channel filter of a wireless communication receiver is realized. In order to satisfy the stringent adjacent channel attenuation specifications of wireless communications standards, the channel filters must have a narrow transition band which results in large number of taps. Digital Advanced Mobile Phone Systems (D-AMPS) receiver design example in [45] is considered to implement the proposed RNS based reconfigurable channel filter. The sampling rate chosen is 34.02 MHz as in [45]. The channel filters extract 30 kHz D-AMPS channels from the input signal after down sampling by a factor of 350. The passband and stopband edges are 30 kHz and 30.5 kHz respectively. The peak passband ripple (PPR) and peak stopband ripple (PSR) specifications are 0.1 dB and -24 dB respectively. The length of the FIR filter, N , is determined using (7.1) [65]

$$N = \frac{-2 * \log_{10}(10.\delta_p.\delta_s)}{3(f_s - f_p)} - 1 \tag{7.1}$$

where δ_p is the PPR, δ_s is the PSR and $(f_s - f_p)$ is the transition band width normalized by the sampling frequency. The filter length (N) obtained is 200.

The D-AMPS channel filter for a coefficient wordlength of 8 bits is implemented. The filter implementation is shown in Fig. 7.4. The product encoders corresponding to each moduli is designed only once and it is reused for the realization of reconfigurable channel filter architecture. The channel filter is synthesized on TSMC 0.18 μ m CMOS standard cell technology. The synthesis results are shown in Table 7.2 along with a comparison of the proposed method with the reconfigurable FIR filter in [80]. The reconfigurable FIR method in [80] uses binary based CSE technique to optimize the multipliers by using addition and shifts. Thus it results in reduced area implementation when compared to the proposed reconfigurable RNS-FIR filter.

Figure 7.4. High-speed and reconfigurable RNS-FIR filter.

As in any RNS filter implementations, a reverse converter is needed to convert the RNS representation back to binary representation in the proposed design. If the

wordlength of the quantized input signal is less than 8 bits then a forward converter to convert the binary input number to RNS is not required. If quantized input has more than 8 bits, it is converted to respective residue value using a forward converter. It can be noted that the forward and reverse converters can have fixed moduli, as the moduli is selected to satisfy the most stringent communication standard. This will increase the speed of reverse converter. The area overhead of the reverse converter is 0.58 mm² and it is included in the results given in Table 7.2. It can be noted that the proposed method offers a speed improvement of 18.75% and power improvement of 8% over [80].

Table 7.2
0.18μm synthesis result of 200 tap D-AMPS channel filter.

	Proposed RNS-FIR filter	Reconfigurable FIR filter [80]
Filter length	200	200
Area (mm ²)	2.35	1.313
Delay (ns)	4.68	5.76
Power (mW)	13.8	15

RNS based FIR filters found in literature are not reconfigurable to achieve different frequency response specifications as required in the case of multi-standard wireless communication receivers. Hence the proposed method is compared with general reconfigurable filters available in literature [55-58]. Reconfigurable filters in [55-58] mostly deal with lower-order filters. A comparison of the proposed method with [55-58] is given in Table 7.3. This comparison shows that the proposed method works well for lower-order filters like the masking filters in Chapter 5 and 6. The synthesis results of the digit reconfigurable method [55] and the high-speed reconfigurable CSD method [56] are taken from [80]. From Table 7.3, it can be noted that the proposed method has the least AT complexity and power when compared with the 32 tap implementations in [55] and [57]. The RNS-FIR method offers a speed improvement of 57.28% over the architecture in [55] and 64.43% over [57] at the cost of an increase in area by 1.13%. The AT complexity of the proposed architecture is 58.95% less than that of [55] and 63.96% less than [57]. The proposed method offers a power savings of 65.8% over [55] and 55.4% over [57]. The MMCM method in [58] is having an area reduction of 61.2% when compared to the proposed method, as the

multiplication is done serially. The delay for MMCM method is 83.65% higher than the proposed reconfigurable RNS-FIR filter and hence the AT complexity of MMCM method is 57.8% higher when compared to the proposed method. The power dissipation of the CSD based reconfigurable FIR in [56] is not reported in [56]. The RNS-FIR method has a huge power savings of 81.8% over [58]. The proposed reconfigurable RNS-FIR method outperforms the high-speed CSD method in [56] in area and delay by 89.8% and, 63.6% respectively.

Table 7.3
0.18μm synthesis result in comparison with reconfigurable FIR filters [55-58].

	RNS-FIR	FIR filter [55]	ReMB filter [56]	CSD-FIR [57]	MMCM - filter [58]
Coefficient wordlength (bits)	8	8	8	10	16
Filter length	32	32	32	18	20
Area (mm ²)	1.41	1.47	1.394	13.872	0.5467
Delay (ns)	2.55	5.97	7.17	7	15.6
Area-Time Complexity (mm ² x ns)	3.6	8.77	9.99	97.1	8.53
Power (mW)	2.9	8.5	6.5	-	16

7.4 Summary

In this chapter, a high-speed reconfigurable FIR filter based on RNS is presented. The results show that the RNS-FIR filters are good for reconfigurable lower-order masking filters described in Chapter 5 and 6 as well as for higher-order channel filters for SDR. The proposed architecture can be reconfigured to satisfy the frequency response specifications of different wireless communication standards for a given moduli set. Due to the parallel nature of RNS arithmetic the architecture could operate at high-speed. The results show that the proposed product encoder offers AT improvements of 67% improvement over [73], 62.2% over [75], 23.43% over [76], 35.8% over [78] and 24.4% over [79]. The synthesis results using TSMC 0.18μm CMOS standard cell technology for D-AMPS channel filter are also presented. The proposed method has a speed improvement of 18.75% over [80]. The AT complexity of the architecture is 58.95% less than that of [55] and 63.96% less than [57]. The proposed method offers a power savings of 65.8% over [55] and 55.4% over [57].

Chapter 8

Conclusions and Future Work

In this chapter, a brief summary of the works accomplished in this thesis is presented. Some topics for future work in this research area are also discussed.

8.1 Conclusions

This thesis addressed the hardware-efficient implementation issues of the reconfigurable channel filters and filter banks in the digital front-end of a software defined radio (SDR) receiver. The channel filter extracts narrowband channels from a wideband input signal, down converts each of these narrowband channels to baseband using digital down converters and reduces the sampling rate of the narrowband channels to the sampling rate of the desired standard using a sample rate converter. The channel filtering or filter bank comes directly after the ADC and thus needs to operate at very high sampling rate and also needs to be implemented with low power to work on battery operated devices. In addition to this, reconfigurability is also a requirement to support the multi-standard operation. Thus, reconfigurable, low power and high-speed channel filters and filter banks are needed for the successful deployment of SDR. In this thesis, new techniques to implement low power reconfigurable digital filter banks with area efficiency and high-speed for multi-standard wireless communication receivers have been proposed.

Digital filters are basic building blocks for any digital filter bank. Hence optimization of digital filter is an essential requirement. FIR filters are widely employed in digital filter banks as they have absolute stability and linear phase property which are essential for wireless communication systems. The most computationally intensive operation in an FIR filter is coefficient multiplication. Since higher-order FIR filters, i.e., filters with several hundreds of coefficients, are required in SDR receivers to meet the stringent adjacent channel attenuation specifications of wireless communication systems, low complexity techniques such as common

subexpression elimination (CSE) algorithms have been employed to reduce the complexity of these digital filters. In CSE algorithms, coefficient multiplications are considered as shift and add operations. The goal of CSE is to identify multiple occurrences of identical bit patterns, called common subexpressions (CSs), that are present in the coefficients, and eliminate redundant multiplications (due to these CSs) to minimize the number of adders needed to realize the coefficient multipliers. Conventionally CSE algorithms use canonical signed digit (CSD) representation filter coefficients because of reduced number of non-zero digits compared to binary representation of filter coefficients. It was reported in [42] that CSD-based CSE techniques have constraints in reducing the number of adders due to the presence of signed bits, which may require additional adders. This constraints of CSD-based CSE methods resulted in a 3-bit binary based CSE method in [47] called Binary Subexpression Elimination (BSE) method, which produced better reduction of adders compared to the CSD-based CSE methods [36]-[46]. However, it has been found that there is an overhead of additional adders due to the delay differences between non-zero terms (Vertical Common Subexpressions and unpaired non-zero bits) of adjacent coefficients, when Vertical common subexpression elimination used in [47], is applied. It was also found that 4-bit binary CSs has the possibility to reduce the number of adders without affecting the logical depth when compared to the best known CSE technique [47] which uses 3-bit binary CSs. Based on these findings, a 4-bit Binary Common Subexpression Elimination method (BCSE-4) was proposed in this thesis. As more number of non-zero bits are grouped in BCSE-4, greater reduction of adders is possible compared to BSE [47]. The reduction of adders achieved using the proposed BCSE-4 method is slightly inferior when compared to CSD based CSE methods for short filters. However, the BCSE-4 method offers better reduction of adders for higher-order filters. Therefore, the proposed BCSE-4 method is best suited for implementing higher-order FIR filters in SDR channelizers. The design examples show that the proposed BCSE-4 method offers an average adder reduction of 29.1% compared to [40], 27.9% compared to [45], 6.5% compared to [47] and 28.4% compared to [51].

The second method incorporates reconfigurability with low complexity for the implementation of a channel filter using multi-band filtering and masking. The proposed channel filter is able to extract channels of multiple communication

standards, with different bandwidths and different stopband attenuations using a fixed-coefficient multi-band filter and appropriate reconfigurable masking filters according to the current mode of operation. As the same multi-band filter is used for different specifications, with filter order proportional to the most stringent communication standard, the over-all complexity can be reduced. The BCSE-4 method can also be applied to the multi-band filter as it is a fixed co-efficient filter. The reconfigurability is achieved by swapping the masking filter coefficients according to current mode of operation using LUT as they are of lower-order nature. The proposed multi-band filtering and masking architecture is implemented on Xilinx Virtex 2v3000ff1152-4 FPGA associated with the dual DSP-FPGA Signalmaster kit provided by Lyrtech [66] using real-time inputs. The method offers complexity reduction of 51.2% and average power reduction of 12.7% when compared to per-channel (PC) approach.

The FIR filters in SDR receivers require sharp transition-bands because of stringent adjacent channel attenuation specifications and thus higher-order in nature. The higher-order FIR filters have large number of multiplications and hence highly complex. A frequency response masking (FRM) technique was employed in [63] to design sharp transition-band FIR filters with low complexity. The basic idea was to compose the over-all FIR filter with sharp transition-band requirements using three wide transition-band FIR filters i.e., one modal filter and two masking filters. The modal filter is a special type of filter, where each unit delay of the normal filter is replaced by L delays or in other words interpolating the frequency response by L . The output of the modal filter consists of $L+1$ frequency bands. Since the subfilters have wide transition-band specifications, resulting filters are of lower-order. However if the desired frequency response has very narrow transition-band width, the modal filter needs to be interpolated using a very high value of L . When L is large, the resulting $(L+1)$ multi-band channels lie closer compared to the case when L is small. Consequently, higher-order masking filters would be required to extract the bands of interest. This will increase the over-all cost of the proposed filters and filter banks. This cost can be reduced by employing multiple stages of the FRM technique. As the third work in this thesis, reconfigurability and extraction of channels with very narrow transition bandwidth, that require larger value of L , is realized using a low complex method of multi-stage FRM (MSFRM) approach. A dynamically reconfigurable filter bank and channel filter has been proposed based on a MSFRM technique. The

proposed channel filter and filter bank has the flexibility of dynamically changing the frequency responses by changing the interpolating values from a very small value to a large one, with very low complexity. The MSFRM approach helps the user to take benefit from large interpolation values, which caters to wide choices of bandwidth and centre frequencies of the multiple standard radio channels. All the filters in MSFRM technique are having wide transition-band for any arbitrary interpolation value and hence they are less complex. Design examples show that the proposed MSFRM based channel filter offers 47.9% reduction in complexity compared to SSFRM based channel filter [67]. The proposed reconfigurable filter bank allows the user to enjoy non-uniform bandwidth extraction, very narrowband channel extraction and multi-standard operation when compared to conventional filter banks in SDR channelizers. The proposed filter bank has been compared with conventional per-channel (PC) approach, DFT filter bank (DFTFB), Goertzel filter bank (GFB) [17] the modulated perfect reconstruction filter bank (MPRFB) [26] and single-stage FRM (SSFRM) approach [67] and the architecture is implemented and tested on Virtex 2v3000ff1152-4 FPGA. The proposed method offers an average gate count reduction of 36.5% and power reduction of 18.1% with a slight increase in delay of 1.5% when compared to SSFRM based architecture [67].

In SDR receivers, the channel filters often need to extract channels, which have bandwidths that are related by non-integer factors. Based on the FRM approach, a new channel filtering method has been proposed, called the decimation-interpolation and masking (DIM) method, which can extract channels, whose bandwidths are related by fractional factors, which forms the fourth contribution. The fractional bandwidth extraction is made possible by using a two-stage approach, where the first stage of the proposed architecture employs a low pass linear phase FIR filter, called the *modal filter*, whose frequency band edges can be varied by changing the decimation and interpolation factors. By varying decimation and interpolation values, fractional bandwidth of the modal filter response can be achieved. Thus the first stage will provide bandwidth adaptation, i.e., tuning of frequency bands corresponding to the channel spacing of a distinct communication standard. Based on the time-varying channels, the desired channel is adapted from the decimated and interpolated filter output using appropriate masking filters at the second stage. The proposed channel filter offers reconfigurability at the architectural level and at the channel filter level

and allows the user to have better control with respect to MSFRM approach, over the bandwidth and centre frequency, for the extraction multiple radio channels. The implementation results show that method offers an average gate count reduction of 10.7% over the PC approach. The average reduction of power consumption achieved using the proposed method over the PC approach is 5.9%. The proposed DIM based channel filter architecture offers 95.7% area-time complexity reduction when compared to the reconfigurable multiplier block filters [57].

As the fifth contribution, new reconfigurable, low complexity FIR filter architecture for realizing masking filters of the MSFRM and DIM based channel filters, based on the residue number system (RNS), has been proposed. This method focuses on direct implementation of RNS-FIR filters using look up tables (LUTs) and thus results in high-speed operation. In contrast to conventional reconfigurable multiplier units in filter architectures, easily reconfigurable LUT based moduli multipliers have been employed in the proposed architecture. The proposed architecture can be reconfigured to satisfy the frequency response specifications of different wireless communication standards, with varying number of taps, for a given moduli set. Due to the parallel nature of RNS arithmetic the architecture could operate at high-speed. To the best of our knowledge, this is the first attempt towards integrating reconfigurability using RNS technique for implementing higher-order FIR filters. The results show that the proposed product encoder offers AT improvements of 67% improvement over [73], 62.2% over [75], 23.43% over [76], 35.8% over [78] and 24.4% over [79]. The synthesis results using TSMC 0.18 μ m CMOS standard cell technology for D-AMPS channel filter are also presented. The proposed method has a speed improvement of 18.75% over [80]. The AT complexity of the architecture is 58.95% less than that of [55] and 63.96% less than [57]. The proposed method offers a power savings of 65.8% over [55] and 55.4% over [57].

For an easy evaluation, Table 8.1 summarizes the different methods proposed and the key results obtained in this thesis.

Table 8.1

Summary of the methods and results accomplished in this thesis.

	Chapters	Methods	Key results
1	Chapter 4	4-bit Binary Common Subexpression Elimination (BCSE-4)	<ol style="list-style-type: none"> 1) The BCSE-4 method is best suited for implementing higher-order FIR filters in SDR channelizers. 2) The BCSE-4 method offers an average adder reduction of 29.1% compared to [40], 27.9% compared to [45], 6.5% compared to [47] and 28.4% compared to [51].
2	Chapter 5	Multi-band Filtering and masking based channel filter	<ol style="list-style-type: none"> 1) The proposed channel filter incorporates reconfigurability with low complexity and can extract channels of multiple communication standards, with different bandwidths and stopband attenuations using a fixed-coefficient multi-band filter and reconfigurable masking filters. 2) The method offers complexity reduction of 51.2% and average power reduction of 12.7% when compared to per-channel (PC) approach.
3	Chapter 5	Multi-stage Frequency Response Masking (MSFRM) based channel filter and filter bank	<ol style="list-style-type: none"> 1) The proposed channel filter and filter bank has the flexibility of dynamically changing the frequency responses by varying the interpolating values from a very small value to a large one, which caters to wide choices of bandwidth and centre frequencies of the multiple standard radio channels. 2) The method offers average gate count reduction of 36.5% and power reduction of 18.1% with a slight increase in delay of 1.5% when compared to single-stage FRM based architecture.
4	Chapter 6	Decimation-Interpolation and Masking (DIM) based channel filter	<ol style="list-style-type: none"> 1) The DIM based method allows extraction of channels with bandwidth related by fractional factors. The proposed channel filter allows the user to have better control with respect to MSFRM approach, over the bandwidth and centre frequency, for the extraction multiple radio channels. 2) The method offers an average gate count reduction of 10.7% and an average reduction in power consumption of 5.9 % over the PC approach.
5	Chapter 7	Residue Number System (RNS) based reconfigurable and high-speed FIR filter	<ol style="list-style-type: none"> 1) The RNS-FIR method focuses on direct implementation of filters using look up tables and thus results in high-speed operation. For a given moduli set, the architecture can be reconfigured for multiple standards with varying number of taps. 2) The AT complexity of the architecture is 58.95% less than that of [55] and 63.96% less than [57]. The proposed method offers a power savings of 65.8% over [55] and 55.4% over [57].

8.2 Future Work

In this dissertation, the challenges in designing low power reconfigurable high-speed and low complexity digital filter banks for SDR receivers have been investigated. Five new techniques to tackle these challenges were proposed. However many optimization possibilities to pursue further research in this area in future have been identified. Some of them are discussed below:

8.2.1 One-shot Design of Reconfigurable Channel filters using FRM Technique

After reviewing various types of FRM techniques, it is interesting to address the possibilities to implement a one-shot design for extracting channel filters for various standards. For realizing the above reconfigurable filter, the number of delays in the modal filter of FRM approach needs to be changed, so as to change passband width of the overall filter. Integer and fractional multiples of extracted channel passband width when compared to modal filter passband width were obtained using previously proposed multi-stage FRM (MSFRM) and decimation-interpolation masking (DIM) method respectively. But in both the methods the delay elements for realizing different communications standards using same modal filter remained as an integer. The next step is to study about fractional delay filters so that the delay of the modal filter can be varied fractionally. The primary aim is to incorporate fractional delay with FRM technique so as to implement a one-shot design for SDR channel filters. This will enable us to realize any bandwidth specification which is the ultimate aim of SDR channel filter.

8.2.2 Channelizers based on Fast Filter Banks

In this dissertation, non-uniform filter banks were discussed based on MSFRM and DIM approach. But it can be noted that the proposed method needs individual masking filter to mask out each channel and thus the total complexity depends upon number of channels extracted. In future, it will be interesting to study on non-uniform filter banks whose complexity is independent of the architecture. A concept known as fast filter

banks (FFB) was proposed in [81] which resulted in sharp frequency response performance and extremely low complexity compared to conventional Frequency Domain Filtering (FDF) implementations. In FFB architecture, the prototype filter in FDF implementations is replaced by a higher-order filter, which will result in an improved frequency response compared to ordinary fast Fourier transforms (FFT) implementations. In order to reduce the complexity of higher-order prototype filter, the filter is implemented as an odd length symmetrical impulse response half-band filter. Hence the prototype filter has the property that its every other coefficient value is zero. Hence the number of non-trivial multipliers is only about one-quarter of its filter length [81]. The architecture of FFB can be investigated to incorporate reconfigurability and to obtain a non-uniform filter bank. The method in [81] closely follows tree structured architecture as in [25] but with much less delay in obtaining output. Also in an FFB, the wideband input is not divided asymmetrically making it more appropriate for faster channel selection compared to conventional FDF approaches.

8.2.3 Optimization of RNS-FIR Filters for Multiple Moduli

In this dissertation, reconfigurable low complexity and high-speed FIR filter architecture based on the residue number system (RNS) is proposed. The proposed system is built on three moduli set $\{2^n-1, 2^n, 2^n+1\}$ system and the wordlength of the filters was fixed taking care of the most stringent radio channel specification. If there is a need to change the moduli set to make it more general, then the entire architecture has to be redefined for the new moduli. Hence as a future work, it will be advantageous to design a reconfigurable RNS-FIR filter which can not only be reconfigured for multiple standards but also to accommodate multiple moduli sets.

Bibliography

- [1] J. Mitola, *Software radio architecture*, John Wiley, 2000.
- [2] E. Buracchini, "The software radio concept," *IEEE Communications Magazine*, vol. 38, no. 9, pp. 138-143, Sep. 2000.
- [3] J. Mitola, "The software radio architecture," *IEEE Communications Magazine*, vol. 33, no. 5, pp. 26-38, May 1995.
- [4] J. Mitola, "Software radio architecture: A mathematical perspective," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 4, Apr. 1999.
- [5] T. Hentschel, M. Henker and G. Fettweis, "The digital front-end of software radio terminals," *IEEE Personal Communications Magazine*, vol. 51, no. 4, pp. 738-748, Jul. 2002.
- [6] S. Srikanteswara, R. C. Palat, J. H. Reed and P. Athanas, "An overview of configurable computing machines for software radio handsets," *IEEE Communications Magazine*, vol. 41, no. 7, pp. 134-141, Jul. 2003.
- [7] Z. Salcic and C. F. Mecklenbrauker, "Software radio - architectural requirements, research and development challenges," *Proceedings of IEEE International Conference on Communication Systems*, vol. 2, pp. 711-716, Nov. 2002.
- [8] <http://focus.ti.com/docs/prod/folders/print/ads5485.html>
- [9] A. Haghighat, "A review on essentials and technical challenges of software defined radio," *Proceedings of military communications conference*, vol. 1, pp. 377-382, Oct. 2002.
- [10] R. Bagheri, A. Mirzaei, M. E. Heidari, S. Chehrazai, L. Minjae, M. Mikhemar, W. K. Tang and A. A. Abidi, "Software-defined radio receiver: dream to reality," *IEEE Communications Magazine*, vol. 44, no. 8, pp. 111-118, Aug. 2006.
- [11] A. A. Abidi, "The path to the software-defined radio receiver," *IEEE Journal of Solid-State Circuits*, vol. 42, no. 5, pp. 954-966, May 2007.
- [12] R. H. Walden, "Analog-to-digital converter survey and analysis," *IEEE Journal of Selected Areas in Communication*, vol. 17, no. 4, pp. 539-550, Apr. 1999.
- [13] R. H. Walden, "Performance trends for analog-to-digital converters," *IEEE Communications Magazine*, vol. 37, no.2, pp. 96-101, Feb. 1999.
- [14] O. A. Mukhanov et al, "A superconductor high resolution ADC," *IEEE Transaction on Applied Superconductivity*, vol. 11, no. 1, Mar. 2001.
- [15] T. Hentschel and G. Fettweis, "Software radio receivers," *CDMA techniques for the third generation mobile systems*, Kluwer Academic Publishers, 1999.
- [16] T. Hentschel and G. Fettweis, "Sample rate conversion for software radio," *IEEE Communications Magazine*, vol. 38, no. 8, pp. 142-150, Aug. 2000.

- [17] T. Hentschel, "Channelization for software defined base-stations," *Annales des Telecommunications*, ISSN 0003-4347, vol. 57, no. 5-6, pp. 386-420, May-June 2002.
- [18] A. Wiesler and F. K. Jondral, "A software radio for second- and third-generation mobile systems," *IEEE Transactions on Vehicular Technology*, vol. 51, no. 4, pp. 738-748, July 2002.
- [19] L. Pucker, "Channelization techniques for software defined radio," in *Proceedings of SDR forum conference*, Nov. 2003.
- [20] A. Mehrotra, GSM system engineering, Artech House, 1997.
- [21] F. Ananasso, G. Chiassarini, E. Del Re, R. Fantacci, D. Rousset and E. Saggese, "A multirate digital multicarrier demodulator: design, implementation, and performance evaluation", *IEEE Journal on Selected Areas in Communications*, vol. 10, no. 8, pp. 1326-1342, Oct. 1992.
- [22] J. G. Proakis and D. G. Manolakis, *Digital signal processing*. Prentice-Hall, Inc., 3rd ed., 1996.
- [23] J. Lillington, "Comparison of wideband channelization architectures," Apr. 2003. <http://www.techonline.com/community/techgroup/dsp/tech paper/33204>
- [24] J. Lillington, "TPFT - Tunable pipelined frequency transform," RF Engines Ltd, Technical Report, Sep. 2002. <http://www.rfel.com/whipapdat.asp>
- [25] A. P. Vinod, E. M-K. Lai, A.B. Premkumar, and C.T.Lau, "A reconfigurable multi-standard channelizer using QMF trees for software radio receivers," *Proceedings of 14th IEEE International Symposium on Personal, Indoor and Mobile Radio Communication*, pp. 119-123, Sep. 2003.
- [26] W. A. Abu-Al-Saud and G. L. Stuber, "Efficient wideband channelizer for software radio systems using modulated PR filter banks," *IEEE Transactions on Signal Processing*, vol. 52, no. 10, pp. 2807-2820, Oct. 2004.
- [27] C. Y. Fung and S. C. Chan, "A multistage filter bank-based channelizer and its multiplier less realization", *Proceedings of IEEE International Symposium on Circuits and Systems*, vol.3, pp. 429-432, May 2002.
- [28] B. Li, L. Ge and J. Zheng, "Efficient dynamic channelizer based on nonuniform filter banks," *European Transactions on Telecommunications*, vol. 19, no. 3, pp. 273-283, Mar. 2007.
- [29] A. Avizienis, "Signed-digit number representations for fast parallel arithmetic," *IRE Transactions on Electronic Computers*, vol. EC- 10, pp. 389-400. 1961.
- [30] G. K. Ma and F. J. Taylor, "Multiplier policies for digital signal processing," *IEEE ASSP Magazine*, vol. 7, no. 1, pp. 6-20, Jan. 1990.
- [31] D. Li, "Minimum number of adders for implementing a multiplier and its application to the design of multiplierless digital filters," *IEEE Trans. on Circuits and Syst.*, vol. 42, no. 7, pp. 453-460, Jul. 1995.
- [32] I. C. Park and H. J. Kang, "Digital filter synthesis based on minimal signed digit representation," *Proc. of DAC*, pp. 468-473, Jun. 2001.

- [33] H. Samueli, "An improved search algorithm for the design of multiplierless FIR filters with powers-of-two coefficients," *IEEE Transactions on Circuits and Systems*, vol. 36, no. 7, pp. 1044-1047, Jul. 1989.
- [34] D. Li, Y. C. Lim, Y. Lian and J. Song, "A polynomial-time algorithm for designing FIR filters with powers-of-two coefficients," *IEEE Transactions on Signal Processing*, vol. 50, no. 8, pp. 1935-1941, Aug. 2002.
- [35] L. Cen, "A hybrid genetic algorithm for the design of FIR filters with SPoT coefficients," *Journal of Signal Processing*, vol. 87, pp. 528-540, Mar. 2007
- [36] M. Potkonjak, M. B. Srivastava, and P. A. Chandrakasan, "Multiple constant multiplications: Efficient and versatile framework and algorithms for exploring common subexpression elimination," *IEEE Trans. on Computer-Aided Design*, vol. 15, no. 2, pp. 151-161, Feb. 1996.
- [37] R. I. Hartley, "Subexpression sharing in filters using canonic signed digit multipliers," *IEEE Trans. on Circuits and Syst. II*, vol. 43, pp. 677-688, Oct. 1996.
- [38] M. Mehendale, S. D. Sherlekar, and G. Vekantesh, "Synthesis of multiplierless FIR filters with minimum number of additions," *Proceedings of the 1995 IEEE/ACM International Conference on Computer-Aided Design*, pp. 668-671, 1995.
- [39] R. Pasko, P. Schaumont, V. Derudder, S. Vernalde, and D. Durackova, "A new algorithm for elimination of common subexpressions," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Syst.*, vol. 18, no. 1, pp. 58-68, Jan. 1999.
- [40] M. M. Peiro, E. I. Boemo and L. Wanhammar, "Design of high-speed multiplierless filters using a nonrecursive signed common subexpression algorithm," *IEEE Trans. on Circuits and Syst. II*, vol. 49, no. 3, pp. 196-203, Mar. 2002.
- [41] C. Y. Yao, H. H. Chen, C. J. Chien and C. T. Hsu, "A novel common-subexpression-elimination method for synthesizing fixed-point FIR filters," *IEEE Trans. on Computer-Aided Design*, vol. 51, no. 11, pp. 2215-2221, Nov. 2004.
- [42] K. S. Yeung and S. C. Chan, "The design and multiplier-less realization of software radio receivers with reduced system delay," *IEEE Trans. on Circuits and Syst. I*, vol. 51, no. 12, pp. 2444- 2459, Dec. 2004.
- [43] N. Boullis and A. Tisserand, "Some optimizations of hardware multiplication by constant matrices," *IEEE Trans. on computer*, vol. 54, no. 10, pp. 1271-1282, Dec. 2005.
- [44] Y. Jang and S. Yang, "Low-power CSD linear phase FIR filter structure using vertical common subexpression," *Electronics letters*, vol. 38, no. 15, pp. 777-779, Jul. 2002.
- [45] A. P. Vinod and E. M-K. Lai, "On the implementation of efficient channel filters for wideband receivers by optimizing common subexpression elimination methods", *IEEE Trans. on Computer-Aided Design for Integrated Circuits and Syst.*, vol. 24, no. 2, pp. 295-304, Feb. 2005.
- [46] A. P. Vinod and E. M-K Lai, "Comparison of the horizontal and the vertical common subexpression elimination methods for realizing digital filters," *Proceedings of IEEE International Symposium on Circuits and Systems*, vol. 1, pp. 496 – 499, May 2005.

- [47] R. Mahesh and A. P. Vinod, "A new common subexpression elimination algorithm for realizing low complexity higher order digital filters," *IEEE Transactions on Computer-Aided Design of Integrated Circuits & Systems*, vol. 27, no. 2, pp. 217-229, Feb. 2008.
- [48] D. R. Bull and D. H. Horrocks, "Primitive operator digital filters," *IEE Proc. Circuits, Systems and Devices*, vol. 138, no. 3, pp. 401-412, Jun. 1991.
- [49] A. G. Dempster and M.D. Macleod, "Use of minimum-adder multiplier blocks in FIR digital filters," *IEEE Trans. Circuits and Syst. II*, vol. 42, no. 9, pp. 569-577, Sep. 1995.
- [50] H. J. Kang and I. C. Park, "FIR filter synthesis algorithms for minimizing the delay and the number of adders," *IEEE Trans. on Circuits and Syst. II*, vol. 48, pp.770-777, Aug. 2001.
- [51] F. Xu, C. H. Chang, and C. C. Jong, "Contention resolution algorithm for common subexpression elimination in digital filter design," *IEEE Trans. on Circuits and Syst. II*, vol. 52, pp.695-700, Oct. 2005.
- [52] Y. Voronenko and M. Pushcel, "Multiplierless Multiple Constant Multiplication," *ACM Transactions on Algorithms*, vol. 3, no. 2, May 2007.
- [53] T. Solla and O. Vainio, "Comparison of programmable FIR filter architectures for low power," in *Proc. of 28th European Solid-State Circuits Conference*, pp. 759-762, Sep. 2002.
- [54] J. Park et al, "Computation Sharing Programmable FIR Filter for low power and high performance," *IEEE Journal on Solid State Circuits*, vol. 39, no. 2, pp. 348-357, Feb. 2004.
- [55] K. H. Chen and T. D. Chiueh, "A low-power digit-based reconfigurable FIR filter," *IEEE Trans on Circuits and Systems-II*, vol. 53, no 8, pp. 617- 621 , Aug. 2006.
- [56] T. Zhangwen and J. Zhang and H. Min, "A high-speed, programmable, CSD coefficient FIR filter," *IEEE Trans. on Consumer Electronics*, vol. 48, pp. 834-837, Nov. 2002.
- [57] S. S. Demirsoy, I. Kale, A. G. Dempster, "Efficient implementation of digital filters using novel reconfigurable multiplier blocks," in *Proceedings of Thirty-Eighth Asilomar Conference on Signals, Systems and Computers*, vol. 1, pp. 461-464, Nov. 2004.
- [58] P. Tummeltshammer, J. C. Hoe and M. Puschel, "Time-multiplexed multiple-constant multiplication," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 9, pp. 1551-1563, Sep. 2007.
- [59] K-Y. Khoo, A. Kwentus and A. N. Willson, "A programmable FIR digital filter using CSD coefficients," *IEEE Journal on Solid State Circuits*, vol. 2, no. 6, pp. 869-874, Jun. 1996.
- [60] Y- J. Oh, H. Lee and C-H. Lee, "A reconfigurable FIR filter design using dynamic partial reconfiguration," *Proceedings of IEEE International Symposium on Circuits and Systems*, pp. 4581 – 4584, May 2006.
- [61] R. A. Walker and D. E. Thomas, *A Survey of High-level Synthesis Systems*. Boston, MA: Kluwer, 1991.
- [62] K. C. Zangi and R. D. Koilpillai, "Software radio issues in cellular base stations," *IEEE Journal on Selected Areas of Communications*, vol. 17, no.4, pp. 561-573, Apr. 1999.

- [63] Y. C. Lim, "Frequency-response masking approach for the synthesis of sharp linear phase digital filters", *IEEE Trans. Circuits. Syst.*, vol.33, pp. 357-364, Apr. 1986.
- [64] <http://www.rfcafe.com/references/electrical/wireless-comm-specs.htm>
- [65] M. Bellanger, "On computational complexity in digital filters," in *Proc. of the European Conference on Circuit Theory and Design*, Hauge, Netherlands, pp. 58-63, Aug. 1981.
- [66] http://www.lyrtech.com/DSP-development/dsp_fpga/signalmaster_quad_cpci.php
- [67] R. Mahesh and A. P. Vinod, "Reconfigurable frequency response masking filters for software radio channelization," *IEEE Transactions on Circuits & Systems II*, vol. 55, no. 3, pp. 274-278, Mar. 2008.
- [68] R. Mahesh and A. P. Vinod, Coefficient decimation approach for realizing reconfigurable finite impulse response filters, in: *Proceedings of the 2008 IEEE International Symposium on Circuits and Systems*, 2008, pp. 81-84.
- [69] N. S. Szabo and R. I. Tanaka, "Residue arithmetic and its applications in computer technology", New York: McGraw-Hill, 1967.
- [70] U. M. Base, A. Garcia, F. Taylor, "Implementation of a communications channelizer using FPGAs and RNS Arithmetic" *Journal of VLSI Signal Processing Systems*, vol. 28 , no. 1-2, pp. 115 – 128, Jun. 2001.
- [71] G. L. Bernocchi, G. C. Cardarilli, Del Re, A. Nannarelli , "A hybrid RNS adaptive filter for channel equalization," *Fortieth Asilomar Conference on Signals, Systems and Computers*, pp. 1706 – 1710, Oct.-Nov. 2006.
- [72] G. C. Cadarilli et al., "Residue number system for reconfigurable datapath," in *Proc. IEEE int. Symp. On Circuits and Syst.*, vol. 2, pp. 756-759, May 2002.
- [73] S. Menon, C. H. Chang, "A reconfigurable multi-modulus modulo multiplier," in *Proc. IEEE Asia pacific conference on Circuits and Systems.*, pp. 1168-1171, Singapore, Dec.2006.
- [74] M. A. Soderstrand, K. Al-Marayati , "VLSI implementation of very-high-order FIR filters," *IEEE International Symposium on Circuits and Systems*, 1995. vol. 2, no. 30, pp. 1436 – 1439, May 1995.
- [75] R. Zimmerman, "Efficient VLSI implementation of modulo (2^n+1) addition and multiplication," in *Proc. 14th IEEE Symp. Computer Arithmetic*, pp. 158-167, Apr. 1999.
- [76] C. Efstathiou., H. T. Vergos, G. Dimitrakopoulos, and Nikolos, 'Efficient diminished-1 modulo $2^n + 1$ multipliers', *IEEE Trans. Comput.*, 2005, vol. 54, no. 4 pp. 491- 496, Apr. 2005.
- [77] H. T Vergos, C. Efstathiou, "Design of efficient modulo $2^{\lceil n/2 \rceil} + 1$ multipliers", *IET Computers & Digital Techniques*, vol. 1, no. 1, pp. 49 – 57, Jan. 2007.
- [78] Z. wang , G.A. Jullien, and W.C.Miller, " An efficient tree architecture for modulo $2^n+ 1$ multiplication," *J. VLSI Signal Processing*, vol. 14, pp. 241-248, Jun. 1996.
- [79] Y. Ma, "A simplified architecture for Modulo (2^n+1) Multiplication," *IEEE transactions on computers*, vol. 47, no.3, pp. 333-337, Mar. 1998.

- [80] R. Mahesh and A. P. Vinod, "An architecture for integrating low complexity and reconfigurability for channel filters in software defined radio receivers," in *Proceedings of IEEE International Symposium on Circuits and Systems*, pp. 2514-2517, May 2007, USA.
- [81] Y. C. Lim and B. Farhang-Boroujeny, "Fast Filter bank (FFB)," *IEEE Trans. Circuits. and Syst. II*, vol. 39, no. 5, May 1992.
- [82] L. Aksoy, E. O. Gunes, E. Costa, P. Flores, J. Monterio, "Effect of number representation on the achievable minimum number of operations in multiple constant multiplications", IEEE workshop on signal processing systems, pp. 424-429, Oct. 2007.
- [83] S. Shahramian, S. P. Voinigescu, A. Chan Carusone, "A 35-GS/s, 4-bit Flash ADC with active data and clock distribution trees," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 6, pp. 1709-1720, June 2009.
- [84] Nam Sung Kim, Todd Austin, David Blaauw, Trevor Mudge, Kriszti Flautner, Jie S. Hu, Mary Jane Irwin, Mahmut Kandemir, Vijaykrishnan Narayanan, "Leakage Current: Moore's Law Meets Static Power," *Computer*, vol. 36, no. 12, pp. 68-75, Dec. 2003.

List of Publications

Refereed Journals

1. **Smitha K. G.** and A. P. Vinod, "A Low Power Reconfigurable Channel Filter using Multi-band and Masking Architecture for Channel Adaptation in Cognitive radio," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E92-A, no. 6, pp. 1424-1432, June 2009.
2. **Smitha K. G.** and A. P. Vinod, "A New Low Power Reconfigurable Decimation-Interpolation and Masking Based Filter Architecture for Channel Adaptation in Cognitive Radio Handsets," *Physical Communication Journal, Elsevier*, Article in Press: doi10.1016/j.phycom.2009.02.003.(Accepted in February 2009)
3. **Smitha K. G.** and A. P. Vinod, "A Low Complexity Reconfigurable Multi-stage Channel Filter Architecture for Resource-constrained Software Radio Handsets," *Journal of Signal Processing Systems for Signal, Image and Video Technology, Springer*, Article in Press: DOI 10.1007/s11265-008-0330-3 (Accepted in January 2009).
4. **Smitha K. G.** and A. P. Vinod, "Low Power Realization and Synthesis of Higher-order FIR Filters using an Improved Common Subexpression Elimination Method," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E91-A, no.11, pp. 3282-3292, November 2008.

International Conferences

1. **Smitha K. G.** and A. P. Vinod, "A Reconfigurable High-speed RNS-FIR Channel Filter for Multi-standard Software Radio Receivers," *Proceedings of IEEE International Conference on Communication Systems, ICCS 2008*, pp: 1354-1358, November 2008.
2. **Smitha K. G.** and A. P. Vinod, "A New Low Complexity Reconfigurable Channel Filter Architecture for Software Radio Handsets," *Proceedings of IEEE International Conference on Communication Systems, ICCS 2008*, pp: 1334-1338, November 2008.
3. **Smitha K. G.** and A. P. Vinod, "A Reconfigurable Low Complexity Architecture for Channel Adaptation in Cognitive Radio," *Proceedings of 19th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC 2008*, pp: 1-5, September 2008.
4. **Smitha K. G.**, R. Mahesh and A. P. Vinod, "A Reconfigurable Multi-Stage Frequency Response Masking Filter Bank Architecture for Software Defined Radio Receivers," *Proceedings of IEEE International Symposium on Circuits and Systems, ISCAS 2008*, pp: 85-88, May 2008.
5. **Smitha K. G.** and A. P. Vinod, "A New Binary Common Subexpression Elimination Method for Implementing Low Complexity FIR filters," *Proceedings of IEEE International Symposium on Circuits and Systems, ISCAS 2007*, pp: 2327-2331, May 2007.

6. **Smitha K. G.**, Hossam A. H. Fahmy and A. P. Vinod "Redundant Adders Consume Less Energy," *Proceedings of IEEE Asia Pacific Conference on Circuits and Systems*, APCCAS 2006, pp: 422-425, December 2006.

Papers Under Review

1. **Smitha K. G.** and A. P. Vinod, "A High-Speed Reconfigurable Frequency Filter for Software Defined Radio Channelization using RNS", *IEEE Transactions on Circuits and Systems-II*, submitted, July 2009.