Design of a low-power, reconfigurable digital front-end for a multimode SDR handset

Navin Michael

2012

Navin, M. (2012). Design of a low-power, reconfigurable digital front-end for a multimode SDR handset. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/48073>

<https://doi.org/10.32657/10356/48073>

Design of a Low-Power, Reconfigurable Digital Front-end for a Multimode SDR Handset

NAVIN MICHAEL
SCHOOL OF COMPUTER ENGINEERING

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

2012

To my parents and teachers.

Acknowledgements

Research often feels like a lonesome enterprise. But there are many, who consciously and subconsciously play the roles of enablers and catalysts. I would like to take this opportunity to thank them all.

My supervisor, Assoc. Prof. Vinod A Prasad, whose soft, firm and reassuring hand has been the guiding force in my research work.

Prof. Christophe Moy, for his constant support and guidance, and for making my stints at Supelec, Rennes, so enjoyable.

Prof. Jacques Palicot, whose sharp observations played a critical role in shaping this thesis.

Nanyang Technological University, for supporting me with the Graduate Research Scholarship, which allowed me to not worry about finances and just focus on the research.

The French Embassy of Singapore, for supporting me with the Merlion PhD grant and for facilitating the research collaboration with the SCEE team at Supelec, Rennes.

Dr. Mahesh Panicker, for helping me take the first baby steps in research.

My friends, Thilaga, Ram, and Prashob for taking my mind off work, and for making me look forward to each day in the lab. I will miss the innumerable discussions we had.

Dr. Smitha Sreekumar, a friend, guide and mentor, without whom the PhD experience wouldn't have been the same.

And last, but not the least, my parents and my teachers, who shaped me into the person I am today.

Abstract

Emerging communication paradigms like opportunistic spectrum access and wireless heterogeneous networks impose a high degree of flexibility on the underlying physical layer hardware of the mobile terminal. A software implementation of the radio baseband algorithms on an instruction set architecture (ISA) offers the highest degree of flexibility and hardware reusability, since the reconfiguration of the terminal for a new standard involves a mere change of software code. However the power efficiencies of ISAs have not scaled to the point where the entire baseband computations of a mobile handset can be performed in software. The computationally intensive signal processing tasks in the radio baseband have to be accelerated in hardware. Dedicated hardware (HW) accelerator cores have a power efficiency which is several orders higher than a software implementation and hence, have been extensively used for accelerating the computationally intensive tasks like channelization and decoding. HW accelerators, however, are inflexible in general and are optimized for a single specification. Incorporating flexibility necessarily incurs both an area and power penalty. The growing need for supporting multiple wireless standards with heterogeneous throughput and mobility requirements in a small form factor mobile handset with a limited silicon area requires the accelerator cores to be flexible and reusable in addition to being power efficient.

The goal of this thesis is to evolve a framework for designing flexible, power efficient and area efficient multistandard channelization accelerators in the digital front-end of software defined radio (SDR) handsets. SDR is a disruptive paradigm for designing flexible, multimode radios, in which a significant portion of the radio signal processing load is pushed towards the digital domain, where one has the advantages of easier reprogrammability and more

mature power optimization strategies. The digital front-end acts as the interface between the analog-to-digital converter and the subsequent digital baseband blocks. It performs the channelization function, which is one of the most computationally intensive kernels in the radio receiver chain. In a flexible radio, the channelization accelerators should be capable of selecting a channel of arbitrary bandwidth and also perform a sample rate conversion (SRC) operation by arbitrary integral or rational factors. They should also provide sufficient attenuation of interferers, blockers and out of band quantization noise, so that the co-channel interference due to aliasing, and the carrier to noise ratio are below the permissible limits for all modes of operation.

In general the area and power overheads of a flexible accelerator are a function of the granularity of hardware reuse across multiple modes. Reusing hardware blocks at coarser levels of granularity offers more opportunities for high level synthesis behavioral optimizations to reduce the area and power. Reusability of coarse grained operators also reduces the amount of reconfiguration data required to parameterize the accelerator. The channelization tasks of channel selection, interferer attenuation and sample rate conversion can be reduced to a set of filtering and decimation operations. Traditional flexible filter acceleration strategies range from the Velcro approach, where multiple standard specific filters are implemented in parallel with no hardware sharing/reuse, to the filter coprocessor approach, which reuses the same set of coarse grained word-level datapath operators. The former method incurs a high area and leakage power overhead, while the latter method incurs a significant dynamic power overhead. The higher leakage power overhead is particularly significant in battery powered mobile devices which spend most of the time in standby mode, where leakage power dominates the overall power consumption.

The filtering and decimation tasks can be efficiently performed by partitioning the decimation load into multiple stages. This allows low order filters to operate at high sampling rates, and high order filters to operate at low

sampling rates. The filter stages in the multistage decimation filter represent a naturally coarse level of granularity for identifying hardware reuse opportunities. However there are several bottlenecks to directly reusing fixed hardwired filter stages across multiple standards. Firstly there are several ways in which the decimation load may be partitioned across multiple stages. Arbitrarily factorizing the decimation factor for each standard would make the task of finding shared decimation stages difficult. Secondly there exists specialized filter structures like the non recursive cascaded integrator comb filter, the halfband filter, and the Farrow structure which are much more efficient than generic FIR decimation stages, but are effective only at certain stages in the decimation chain, or for certain decimation factors. Hence an efficient and flexible channelization strategy should seek to minimize the computational load by maximizing the use of specialized filter stages, and identify opportunities for hardwiring and reusing filter stages across multiple standards.

The first contribution of this thesis is an analysis of the area, power and reconfiguration latency overheads incurred, when flexibility is incorporated in an accelerator. These overheads are referred to as the flexibility penalty in this thesis. The factors that affect the flexibility penalty components and their impact in a practical energy constrained, resource constrained mobile radio is analyzed. The above components are a function of the granularity of hardware reuse across different modes. We have analyzed the flexibility penalty of traditional filter accelerator designs with respect to the granularity of hardware reuse in the design.

The second contribution of this thesis is a sample rate conversion factorization strategy, that allows arbitrary integral or rational decimation load to be partitioned into fixed regular form. This allows the band edge and stopband attenuation of all the filter stages prior to the last stage to be modified in a manner which allows them to be hardwired and reused across multiple standards with very low reconfiguration overheads. Since it reuses hardware at a much coarser level of granularity than filter coprocessors, it

has a much lower dynamic power consumption and a much lower area overhead than a Velcro style accelerator. Experimental synthesis results have been used to demonstrate the area and power advantage of the proposed accelerator over traditional designs. The runtime parameterizations issues and clocking mechanisms for the proposed channelization accelerator have also been analyzed.

The third contribution of this thesis is a strategy for reducing the power penalty of the filter stages than need to be reprogrammable in the proposed architecture. The need for generic MAC units in programmable filters results in a much higher dynamic power consumption, when compared to the hardwired filter stages. In this thesis, a class of programmable time-shared FIR filters based on fast filter algorithms is proposed, that can trade area for increased timing slack more efficiently than traditional direct form structure based time-shared filters, with increasing levels of parallelism. The increased timing slack in the proposed structure provides more room for supply and threshold voltage scaling, which in turn can be used to reduce all the major nanoscale CMOS power consumption components.

Publications

International Journals

- [1] **Navin Michael**, A. P. Vinod, Christophe Moy and Jacques Palicot, "Design of Multistandard Channelization Accelerators for Software Defined Radio Handsets," *IEEE Transactions on Signal Processing*, vol.59, no. 10, pp. 4767-4780, Oct 2011.
- [1] **Navin Michael**, A. P. Vinod, Christophe Moy and Jacques Palicot, "Flexibility and reusability in the digital front-end of cognitive radio terminals," *Circuits, Systems and Signal Processing Journal, Springer*, vol. 30, no. 4, pp. 799-821, May 2011.
- [2] **Navin Michael**, Christophe Moy, A. P. Vinod and Jacques Palicot, "Area-Power tradeoffs for flexible filtering in green radios," *Journal of Communications and Networks*, vol.12, no.2, pp. 158-167, April 2010.

International Conferences

- [1] Christophe Moy, Wassim Jouini, **Navin Michael**, "Cognitive Radio Equipments Supporting Spectrum Agility," *International Workshop on Cognitive Radio and Advanced Spectrum Management (CogART 2010)*, pp. 1-5, Rome, 7-10 November 2010.
- [2] **Navin Michael**, A. P. Vinod, Christophe Moy and Jacques Palicot, "Low power, flexible FIR filters in the digital front-end of green radios," *Proceedings of IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, pp. 2787-2792, Istanbul, September 2010.
- [3] **Navin Michael**, A. P. Vinod, Christophe Moy and Jacques Palicot, "Area-efficient time-shared FIR filters in nanoscale CMOS," *Proceedings of IEEE International Conference on Green Circuits and Systems*, pp. 54-59, Shanghai, June 2010.
- [4] **Navin Michael**, A. P. Vinod, Christophe Moy and Jacques Palicot, "Design paradigm for standard agnostic channelization in flexible mobile radios," *Proceedings of IEEE International Symposium on Circuits and Systems*, pp. 65-68, Paris, May-June 2010.

- [5] **Navin Michael**, A. P. Vinod, Christophe Moy and Jacques Palicot, "Design of low power multimode time-shared filters," *Proceedings of 7th IEEE International Conference on Information, Communications and Signal Processing*, pp. 1-5, Macau, December 2009.
- [6] **Navin Michael** and A. P. Vinod, "Reconfigurable architecture for arbitrary sample rate conversion in software defined radios," *Proceedings of 19th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, pp. 1-6, Cannes, September 2008.

Contents

List of Figures	ii
List of Tables	iii
List of Abbreviations	iv
1 Introduction	1
1.1 Background	1
1.2 Motivation	5
1.3 Objectives and Contributions	8
1.4 Organization	11
2 The Digital Front-end of Software Defined Radios	12
2.1 Analog Front-end Architectures	12
2.1.1 Superheterodyne Receiver	13
2.1.2 Zero-IF Receiver	14
2.1.3 Low-IF Receiver	14
2.1.4 Wideband-IF Receiver	15
2.2 Digitization in a Multistandard Radio	16
2.3 The Digital Front-end Functions	19
2.3.1 Digital Downconversion	21
2.3.2 Pulse Shaping	22
2.3.3 Sample Rate Conversion	25
2.3.4 Interferer Attenuation	27
2.4 Specialized Filter Structures For Channelization	28
2.4.1 Cascaded Integrated Comb (CIC) Filter	29

2.4.2	Farrow Structure	33
2.4.3	Halfband Filter	35
2.5	Implementation of the DFE in existing SDR baseband architectures . .	37
2.6	Limitations of Existing Decimation Filter Designs for SDR Channelization	43
2.7	Conclusion	48
3	Flexibility Penalty and Hardware Reuse in Flexible Filter Accelerators	49
3.1	The Flexibility Penalty	52
3.1.1	Power Consumption	52
3.1.1.1	Subthreshold Leakage Power	54
3.1.1.2	Gate Leakage Power	56
3.1.1.3	Dynamic Power	56
3.1.2	Area	57
3.1.3	Hardware Reconfiguration Latency	58
3.2	Design Paradigms for Flexible Filter Accelerators	61
3.2.1	Velcro Approach	61
3.2.2	Reconfigurable Datapath with Coarse Grained Functional Units	63
3.2.3	Fine Grained Reconfigurable Fabric with Lookup Table based Constant Multipliers	65
3.2.4	Filter Coprocessor	67
3.2.5	Reconfigurable Constant Multiplier based Programmable Filter .	68
3.3	Summary	70
4	Design of Multistandard Channelization Accelerators for Software Defined Radio Handsets	73
4.1	Introduction	73
4.2	Proposed Multistandard Channelization Accelerator Design Flow	75
4.2.1	A Mathematical Formulation of the Multistandard Channeliza- tion Problem	75
4.2.2	Eliminating the Dependency of Filter Stages on Standard Specific Parameters	78
4.2.3	Fixed Factorization Method for Multistandard Channelization .	81
4.3	Practical Filter Design Considerations	86

4.3.1	Design Considerations for NRCIC Stage	86
4.3.2	Design Considerations for Fractional Decimation	87
4.3.3	Partitioning of the Decimation Load	89
4.4	Experimental Synthesis Results	90
4.5	Clocking and Runtime Parametrization Issues in the Multistandard Channelization Accelerator	99
4.5.1	Clocking	99
4.5.2	Runtime Parametrization and the Reconfiguration Bitstream . .	103
4.6	Summary	106
5	Design of Power Efficient Programmable Time-shared Filters	109
5.1	Parallelism and Power Consumption in Nanoscale CMOS Circuits . . .	111
5.1.1	Trading area for timing slack	111
5.1.2	Interplay between architectural and device level power optimizations	113
5.2	Area Efficiency of Time-shared FIR Filters	116
5.2.1	Direct Form Structure based Time-shared Filters	116
5.2.2	FFA based Time-shared Filters	117
5.2.2.1	Fast Filter Algorithms	117
5.2.2.2	Proposed Class of FFA based Time-shared Structures .	120
5.2.3	Comparison of Area Efficiencies	122
5.2.4	Practical Limits for the Use of FFA Structures	123
5.3	Summary	128
6	Conclusion and Future work	130
6.1	Conclusion	130
6.2	Future Work	133
6.2.1	Optimum Supply/Threshold Voltage Scaling for Minimizing the Total Power Consumption	133
6.2.2	Flexible and Reusable Common Operators in Frequency-domain Implementation of the Channelization Function	134
	Bibliography	136

List of Figures

1.1	Analog/Digital boundary in the wireless transceiver chain.	4
2.1	Superheterodyne receiver.	13
2.2	Zero-IF receiver.	14
2.3	Low-IF receiver.	15
2.4	Wideband-IF receiver.	15
2.5	ADC input in the fixed digitization bandwidth approach for different standards.	17
2.6	GSM Interferers.	18
2.7	GSM Blockers.	18
2.8	Distribution of Quantization Noise in Nyquist ADCs.	19
2.9	Distribution of Quantization Noise in $\Sigma\Delta$ ADC.	19
2.10	Spectrum sensing over two bands with different channel bandwidths. . .	21
2.11	Digital downconversion of a complex input signal.	22
2.12	Pulse shaping in the receiver chain.	24
2.13	Input and output sample instants for arbitrary sample rate conversion. .	26
2.14	Cascaded Integrator Comb Structure.	30
2.15	Nonrecursive CIC structure.	31
2.16	Polyphase implementation of the term $\left[\frac{1+z^{-1}}{2}\right]^5$	31
2.17	CIC filter magnitude response.	32
2.18	Farrow structure.	34
2.19	Division of baseband load between 4 cores in LeoCore Processor	38
2.20	Embedded Vector Processor	39
2.21	MUSIC Processor	39
2.22	SODA processor	40

2.23	Montium tile	41
2.24	ADRES baseband core	42
3.1	Single mode ASIC channelization accelerator design flow.	50
3.2	The flexible accelerator design space and the flexibility penalty.	51
3.3	CMOS power consumption trends.	53
3.4	Leakage currents in a MOSFET.	54
3.5	Periodic sensing structure.	61
3.6	Velcro style accelerator.	61
3.7	Synthesis of Multimode ASICs using Graph Merging.	63
3.8	Portion of a Filter CDFG.	64
3.9	CDFG merging illustrated for three constant integer multipliers (19, 41 and 10).	65
3.10	16x16 KCM Multiplier.	65
3.11	p-tap (pipeline registers not shown).	69
3.12	BCSE based Reconfigurable Filter Tap.	70
4.1	Position of alias intervals.	76
4.2	Ideal decimation filter stage for factor ‘p’.	79
4.3	Feasible decimation filter stage for factor ‘p’.	79
4.4	Position of $H_1(z)$ and $H_2(z)$ in the decimation filter chain.	80
4.5	Frequency responses of $H_1(z)$, $H_2(z)$ and $H_{12}(z)$	81
4.6	Programmable nonrecursive CIC structure.	84
4.7	Transpose Farrow structure.	84
4.8	CIC droop correction.	86
4.9	Frequency response of a transpose Farrow Structure.	89
4.10	Proposed multistandard decimation filter.	93
4.11	Pulse output DDS.	100
4.12	Clock enable counter.	101
4.13	Clock enable counters for the proposed multistandard channelization ac- celerator.	102
4.14	Tapped delay line based configuration memory.	104
4.15	The channelization accelerator architecture.	105
4.16	The reconfiguration bitstream.	105

4.17	The control flow for parameterizing the accelerator	106
4.18	Loading of the tapped delay line based configuration memory	107
5.1	Traditional supply/threshold voltage scaling.	113
5.2	Locus of (V_{th}, V_{DD}) points having a fixed frequency constraint.	114
5.3	Voltage scaling for nanoscale CMOS technologies.	114
5.4	Normalized delay as a function of the threshold voltage.	117
5.5	Direct form structure based time-shared filter.	118
5.6	2x2 FFA.	119
5.7	4x4 FFA.	120
5.8	2x2 2 FFA.	121
5.9	Area of time-shared filters whose MAC units operate at $Nf_{clk}/16$	126
5.10	Area of time-shared filters whose MAC units operate at $Nf_{clk}/32$	127
5.11	$N_{K \times K L}$ vs. L plot for different FFA structures.	128

List of Tables

2.1	Pulse shaping parameters	25
2.2	Properties of the Specialized filter structures	36
2.3	DFE Implementation in SDR Baseband Architectures	43
3.1	Reconfiguration data and flexibility (Number of modes supported is assumed to be N , Coefficient wordlength is assumed to be 16 bit)	71
4.1	Channelization Parameters	91
4.2	Factorization of SRC Factor	92
4.3	Gate count of different channelization accelerator implementations	94
4.4	Channelization Computational Load	95
4.5	Dynamic power consumption in different modes.	96
4.6	Factorization of SRC Factor for LTE	97
4.7	Energy of MAC operation of three alternative implementation styles	98
4.8	Dynamic power consumption in different modes.	98
5.1	Dynamic power consumption in different modes.	110
5.2	Number of subfilters and preprocessing/postprocessing adders for different FFA structures	119
5.3	FFA Parameters.	123
5.4	MAC, Adder and Register Area	125

List of Abbreviations

ADC	Analog-to-Digital Converter
ASIC	Application Specific Integrated Circuit
BCSE	Binary Common Subexpression Elimination
BT	Bandwidth Time Product
CDFG	Control Data Flow Graph
CDMA	Code Division Multiple Access
CIC	Cascaded Integrator Comb
CLB	Configurable Logic Blocks
CMOS	Complementary Metal Oxide Semiconductor
CSD	Canonical Signed Digit
CSE	Common Subexpression Elimination
C/N	Carrier to Noise Ratio
CR	Cognitive Radio
DBIL	Drain-induced Barrier Lowering
DC	0 Hz
DDS	Direct Digital Synthesizer
DECT	Digital Enhanced Cordless Communications
DFE	Digital Front-end
DFG	Data Flow Graph
DF	Direct Form
DSP	Digital Signal Processor
FFA	Fast Filter Algorithm
FIR	Finite Impulse Response
FPGA	Field Programmable Gate Array
GOPS	Giga Operations Per Second

GSPS	Giga Samples Per Second
Gbps	Giga Bits Per Second
GSM	Global System for Mobile Communications
HW	Hardware
ICAP	Internal Configuration Access Port
I/O	Input Output
IP	Internet Protocol, Intellectual Property
ISA	Instruction Set Architecture
ISI	Inter Symbol Interference
KCM	Constant Coefficient Multiplier
LCM	Least Common Multiple
LTE	Long Term Evolution
LUT	Look-up Table
MAC	Multiply and Accumulate
MPSoC	Multiprocessor System on Chip
NRCIC	Non-recursive Cascaded Integrator Comb
OSR	Oversampling Ratio
PLL	Phase-Locked Loop
RAM	Random Access Memory
RF	Radio Frequency
RISC	Reduced Instruction Set Computer
RC	Raised Cosine
RRC	Root Raised Cosine
SIMD	Single Instruction Multiple Data
SDR	Software Defined Radio
SRAM	Static Random Access Memory
SRC	Sample Rate Conversion
TOPS	Tera Operations Per Second
TCXO	Temperature Compensated Crystal Oscillator
UMTS	Universal Mobile Telecommunications System
W-CDMA	Wideband Code Division Multiple Access
WiMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Network

WPAN Wireless Personal Area Network

1

Introduction

1.1 Background

The first portable cellular phones offering voice telephony were introduced in the early eighties. Since then the wireless industry has experienced an exponential growth, which can be attributed to the overwhelming advantages of productivity and mobility offered by wireless communications. Wireless technologies have evolved from a few voice only services to a multitude of standards, catering to a wide variety of functional niches. A typical modern consumer might access a cellular network like Global System for Mobile Communications (GSM) or Universal Mobile Telecommunications System (UMTS) for voice telephony, a wireless local area network like IEEE 802.11b or a wireless broadband network like Worldwide Interoperability for Microwave Access (WiMAX) for data connectivity, a mobile multimedia broadcast service like Digital Video Broadcasting-Handheld (DVB-H) and a wireless personal area network (WPAN) like Bluetooth for connecting various peripherals. The ideal scenario from a wireless device manufacturer's point of view would be to have a single standard which can cater to all the consumer's wireless connectivity needs. However the heterogeneous throughput and mobility requirements for providing multiple services like voice telephony, video/audio broadcast and Internet connectivity have resulted in heterogeneous standards, providing varying tradeoffs between throughputs, mobility, coverage and power consumption [1]. While a true convergence of the existing standards to an unified standard might not be feasible or even desirable, one of the important goals envisaged for 4G networks is the decoupling of user service from the access network. The evolution to 4G envisions

1.1 Background

an all-IP core network with standard agnostic terminals providing seamless mobility across multiple, heterogeneous access networks. This can be used to support an ‘always best connected’ model, where the choice of the access network can be matched to the quality of service (QoS) requirement and power consumption constraints [2]. Hence in a low mobility scenario, an user can seamlessly switch from a cellular network to a local area network, which offers lower mobility but a much higher throughput. Such a handover may also be used for offloading the congestion in the cellular network. Designing such a seamless heterogeneous network presents significant network level challenges. An equally challenging issue is the problem of providing flexibility and heterogeneity in a resource and energy constrained mobile handset. In addition to 4G, another emerging communication paradigm that imposes the requirement of flexibility on the mobile handset is opportunistic spectrum access [3]. The exponential increase in the number of wireless users, and the scarcity of wireless spectrum have prompted a relook at the way spectrum is allocated and used. The traditional strategy of licensing spectrum bands often results in inefficient spectrum usage as the bands are out of bounds for unlicensed users, even during the periods when the licensed users are inactive. Monitoring a wide swath of spectrum over a period of time typically reveals multiple vacant slots in the time, frequency and code domains. These vacant slots can potentially be used in an opportunistic fashion for improving the system wide spectrum efficiency. Cognitive radios (CRs) are increasingly being seen as a solution for making opportunistic use of the spectrum [4]. A CR is a radio that is aware of its electromagnetic environment. This awareness can be used for identifying vacant slots for transmission and also relinquishing the slot when a licensed user is detected. Achieving this vision in practice requires a flexible and frequency agile radio that can tune into channels of arbitrary bandwidths over a wide range of frequencies.

The need for supporting multiple standards initially arose in military radios where there is a need for concurrently supporting a large number of transmit and receive channels. Implementing multiple transceiver chains in parallel generally results in bulky and power hungry radios. The software radio was a disruptive radio design paradigm in which the entire radio signal processing load is performed in software by powerful DSPs, after shifting the analog-to-digital divide closer to the antenna as shown in Fig. 1.1 [5, 6]. This allows arbitrary waveforms to be supported by a mere change of software code. Theoretically this offered the ultimate degree of reconfigurability, as switching between

1.1 Background

multiple modes of operation involved a mere change of software. Such a radio could also support in-field upgradeability, since a new standard could be implemented with a simple over-the-air software update. But there are several technological limitations that have prevented the software radio from being a more widespread paradigm. Direct digitization at the RF imposes a prohibitively high sampling rate and dynamic range requirements on the analog-to-digital converter (ADC). It has been estimated that an ADC capable of digitizing the frequency band from 800 MHz to 5.5 GHz, which covers all the present cellular and local area network standards, would require a 12 bit 11 GSPS ADC [7]. The state of art commercially available 12 bit ADCs have reached speeds of 3.6 GSPS [8]. With the above ADC, the analog-to-digital conversion alone would incur a power penalty of about 4.4 W, which is much higher than the 3 W power budget of a typical battery powered mobile handset. Direct RF digitization offers the flexibility to select any number of channels of arbitrary channel bandwidth within the digitized wideband signal. However such a degree of flexibility is an overkill for a mobile handset which typically needs to receive only one channel of interest. Only some specific usage scenarios might require the handset to receive more than one channel at the same time. For instance, an user might be making a voice call over a cellular network while downloading data in the background over a wireless local area network. An user could be performing both the above tasks while using a Bluetooth headset, which would necessitate the simultaneous reception of three channels. In either case, it can be seen that digitizing the entire 800MHz to 5.5GHz band for selecting such a small number of channels is superfluous and the incurred power penalty is too high for a mobile handset. Alleviating the dynamic range and sampling rate requirements of the ADC requires some amount of analog preprocessing and conditioning of the signal before digitization. Thus, a more pragmatic solution would be to limit the radio front-end's mandate to downconverting and selecting a channel of arbitrary bandwidth over a wide frequency band (say for instance, the 800 MHz to 5.5 GHz band covering most of the cellular and WLAN standards). Pushing the analog-to-digital divide towards the antenna is no longer pursued as a critical research goal, instead the focus has largely shifted to designing a physical layer that is flexible and reconfigurable. The software defined radio (SDR) is such a pragmatic radio implementation that incorporates the core advantages of flexibility and in-field upgradeability of Mitola's software radio [7]. While there is still no wide ranging consensus on a rigid definition of SDR, the term has

1.1 Background

come to encompass a wide range of multistandard radio design paradigms which use a flexible hardware platform for performing the physical layer signal processing tasks, and allow various radio parameters like channel bandwidth, carrier frequency, modulation format, blocker requirements etc. to be reconfigured under software control [6].

Fig. 1.1.

Figure 1.1: Analog/Digital boundary in the wireless transceiver chain.

The SDR paradigm presents significant challenges due to the difficulty of supporting flexibility and reprogrammability in both the analog and digital domain. While in general it is easier to provide flexibility in the digital domain, merely pushing the radio signal processing functions to the digital domain cannot guarantee flexibility and reprogrammability. The targeted data rate of 1 Gbps in 4G systems demands much higher channel bandwidths and a dramatically increased computational complexity when compared to 2G and 3G systems. Achieving the spectral efficiency demanded by 4G systems may require a baseband computational load of nearly 5.6 tera ops (TOPS) for a 100 MHz wide channel [2]. Among the existing digital implementation platforms, digital signal processors (DSPs) offer the maximum amount of flexibility as the layer of abstraction offered by the instruction set allows any new algorithm to be implemented with a mere change of software code. However traditional DSPs have a power efficiency of only 1-10 GOPS/W [9]. This implies that the computationally intensive kernels may need to be accelerated in hardware (HW). Application specific integrated circuit (ASIC) HW accelerators have a very high power efficiency, however they are typically hardwired and optimized for a single standard and cannot be readily reused or reprogrammed

for a new specification. Hence, the design of HW accelerators that are flexible as well as power efficient is one of the most important challenges in the digital baseband of a SDR.

Most of the modern smartphones already provide connectivity across multiple access networks, typically one or more cellular networks, a wireless local area network and a wireless personal area network. Hence they could be classified as multistandard radios. However in these devices, the radios corresponding to different standards are typically stacked in parallel with little reuse of hardware. To make the distinction between present-day smartphones and multistandard SDRs more clear, it would be useful to introduce the notions of ‘fixed reconfigurability’ and ‘open reconfigurability’. Fixed reconfigurability refers to the ability of the underlying radio architecture to switch between a set of standards whose specifications are known a priori. This is the level of reconfigurability that is offered by current state of art mobile terminals, some of the common examples being a dual-mode GSM/CDMA phone or a dual-mode GSM/DECT phone. However the notion of open reconfigurability envisions a radio architecture that is ‘standard independent’: not only should the architecture be capable of switching between a known set of standards, it should also offer the flexibility and upgradeability to support a new standard. The SDR methodology offers the promise of achieving such a vision.

SDR has been an active area of research both in the academia and the industry over the past decade. A number of SDR baseband processor designs capable of multistandard support have been reported. Some of the noteworthy designs include IMEC’s ADRES processor [10], Infineon’s MUSIC processor [11], CoreSonic’s LeoCore processor [12], University of Michigan’s SODA processor and its commercial variant designed by ARM called ARDBERG [13], Sandbridge’s Sandblaster processor [14], NXP’s Embedded Vector Processor (EVP) [15], and Recore System’s Montium processor [16], Toshiba’s MT-ADRES [17] and Intel’s Scalable Communications Core [18]. A good overview of most of the current SDR baseband processor designs can be found in [19].

1.2 Motivation

The work presented in this thesis focuses on the design and implementation of the digital front-end (DFE) of a multistandard SDR handset. The DFE is one of the most

computationally intensive blocks in the radio receiver chain. It acts as the interface between the ADC and the subsequent symbol/chip rate baseband processing blocks [20]. The main task of the DFE is to perform the channelization function, which refers to the task of extracting the channel of interest from the digitized ADC output, which may comprise of the required channel along with unwanted interferer and blocker channels. Channelization in a mobile handset is fundamentally different from channelization in a base-station. A base-station typically needs to extract and demodulate multiple channels in parallel. A handset typically has only a single channel of interest. While the latter represents an easier task, the extremely high computational load of the channelization function necessitates the use of power efficient strategies to implement it in an energy-constrained, battery-powered mobile terminal.

The unwanted channels in the DFE need to be attenuated to the noise floor for successful demodulation. The demodulator in the SDR baseband has to operate at the standard specific symbol rate. However the ADC sampling clock may be independent of the above symbol rate. Hence the DFE also needs to perform a sample rate conversion (SRC) operation to decimate the ADC output to the symbol rate. If the channel of interest is at a frequency offset from DC, a digital downconversion stage might be needed to bring the channel to DC. When the pulse shaping load is split between the transmit and receive sides, the DFE also needs to perform a pulse shaping function. Thus the main signal processing tasks that need to be performed in the DFE can be summarized as follows:

- Channel filtering
- Interferer attenuation
- Digital downconversion
- Sample rate conversion
- Pulse shaping

The functionally different channelization tasks can be decomposed into a set of finite impulse response (FIR) filtering and decimation operations. FIR filters are generally preferred in the DFE due to the ease of providing a linear phase and their stability. Filtering involves a large number of computationally expensive multiply and accumulate

operations. There are a few trends which have increased the computational load of the digital frontend. The need for higher data rates as 2G and 3G systems evolve towards 4G requires the channel bandwidths to be increased as it is difficult to significantly increase the spectral efficiency beyond that of present-day standards like LTE and WiMAX [2]. Increased channel bandwidths translate to higher sample rates and an increased computational load. Sigma Delta ($\Sigma\Delta$) ADCs have been emerging as a promising option for performing the analog-to-digital conversion in mobile handsets, due their ability to offer a high resolution within the channel of interest and their lower power consumption [21]. They work on the principle of noise shaping and oversampling. Hence their sample rates are typically much higher than the symbol rate of the standard, resulting in increased computational loads. The channel selection load was traditionally split between the analog and digital filtering stages. Due to the limited flexibility of the analog filters, only a coarse band is selected in the analog domain and a bulk of the channelization load is performed in the digital domain. All these trends have resulted in an increase in the stringency of the digital channel selection filters.

The high computational load, the low-power constraints of mobile handsets, and the regularity of FIR filters make the channelization function a good candidate for hardware acceleration. There is a large body of literature on low-power and low-area HW implementations of constant-coefficient FIR filters [22, 23, 25, 26]. However a multistandard SDR needs to be flexible enough to accommodate multiple channel bandwidths, interference attenuation requirements, pulse shaping factors and sample rate conversion factors. The need for providing multistandard support has now raised questions about a flexibility vs. power consumption/area tradeoff that was never an issue in the case of single mode radios. The programmability requirement necessitates the use of generic hardware, which precludes the use of standard specific constant coefficient filter optimizations. Incorporating flexibility inevitably incurs penalties in terms of area and power. Hence, implementing a low-power multistandard DFE that is also scalable for supporting an arbitrary number of standards requires methodologies to design flexible channelization accelerators whose area and power penalties are not significantly higher than that of single mode channelization accelerators.

1.3 Objectives and Contributions

This thesis seeks to develop a framework for designing flexible, power efficient and area efficient multistandard channelization accelerators in the digital front-end of SDR handsets. The accelerators should be capable of selecting a channel of arbitrary bandwidth and band-edge specifications. The input to the demodulator needs to be at the symbol rate or an integer multiple of the symbol rate. Since the ADC sample rate and the symbol rate can vary across different standards, the channelization accelerator also performs a sample rate conversion operation by arbitrary integral or rational factors. They should also provide sufficient attenuation of interferers, blockers and out of band quantization noise, so that the co-channel interference due to aliasing and the carrier to noise ratio are below the permissible limits for all modes of operation. A limited power budget and a limited silicon area are the overarching design constraints in a mobile handset. Emerging applications like cognitive radios have a very high flexibility requirement. The time taken to reconfigure the physical hardware for operation in a different vacant band reduces the time available for useful communication, which in turn reduces the achievable throughput. Hence the reconfiguration latency of the hardware accelerators is also an important constraint in reconfigurable radios. The work presented in this thesis seeks to reduce the area, power and reconfiguration latency penalty incurred when variable channelization parameters need to be supported by the channelization accelerator. All the functionally different channelization tasks can be performed by a multistandard decimation filter. The Velcro approach [6] and the coprocessor approach [27] represent two ends of the traditional multistandard filter accelerator design space. A Velcro accelerator consists of multiple single-mode accelerators stacked in parallel, each of which have been optimized for a single mode. The Velcro accelerator has the advantage of having a dynamic power consumption which is comparable to that of a single mode accelerator. A spatial style, full parallel implementation of each of the constituent single mode accelerators can make use of constant coefficient filter optimizations which allow the coefficient multiplications to be performed by a network of hardwired shifts and adders. The Velcro approach incurs a significant area penalty as multiple single mode accelerators have to be implemented in parallel. Hence this method is not scalable when a large number of standards need to be supported. Its flexibility is also limited to switching between a known set of standards. This precludes

1.3 Objectives and Contributions

the support for in-field upgradeability which is one of the important goals envisioned for SDRs. The filter coprocessor approach implements the filtering operations in a temporal fashion by mapping the filtering function onto a set of generic multiply and accumulate (MAC) units. The presence of generic MAC units allows any arbitrary coefficient set to be mapped to the same underlying hardware. However its flexibility comes at the expense of a significant dynamic power consumption penalty when compared to single mode accelerators. Hence, both the above accelerator paradigms falls short of the dual requirements of flexibility and power efficiency of a SDR handset. The objective of this thesis is to evolve an alternate channelization accelerator paradigm that has the following properties.

- Lower dynamic power consumption than the filter coprocessor approach
- Lower reconfiguration overhead than the filter coprocessor approach
- Lower area overhead than than the Velcro approach
- Flexibility to switch between a set of standards
- Support for in-field upgradeability

The contributions of this thesis can be clubbed into three broad categories. Firstly, we have introduced the notion of the flexibility penalty for flexible accelerators, analyzed its practical repercussions, and studied the flexibility penalty of traditional flexible filter accelerator designs. Secondly, we have proposed strategies for reducing the flexibility requirement of a multistandard channelization accelerator by identifying opportunities for reusing fixed hardwired hardware blocks across multiple standards with low reconfiguration overheads. Thirdly, we have proposed strategies for minimizing the area and power penalty of the portions of the channelization accelerator that need to be fully reprogrammable. The contributions are elaborated below.

1. Analysis of the flexibility penalty in flexible filter accelerators

A single mode accelerator benefits from a large class of high level synthesis behavioral optimizations, which reduce its area an power. Incorporating flexibility into the accelerator precludes the use of these optimizations. Additionally a flexible accelerator may also need a large amount of reconfiguration data to parameterize it for a new mode, which results in a reconfiguration latency. We refer to these

1.3 Objectives and Contributions

additional area, power and reconfiguration overheads of a flexible accelerator as the flexibility penalty. The contributing factors and the practical repercussions of the flexibility penalty in an energy constrained and resource constrained flexible mobile terminal are analyzed. The area, power and reconfiguration overheads are all closely related to the degree of hardware reuse across multiple modes. We have studied the flexibility penalty of traditional flexible filter accelerator designs, and related it to the granularity of hardware reuse in the particular design style.

2. A standard agnostic channelization accelerator design paradigm

Reducing the area, power and reconfiguration latency penalties requires the reuse of hardware at a very coarse level of granularity. This allows the use of high level synthesis optimizations for reducing the area and power overheads of the hardwired reused portions of the accelerator, while reducing the computational load assigned to generic operators. The principal channelization tasks of filtering and decimation by a large sample rate conversion factor can be efficiently performed by a multistage decimation filter. The filter stages in the above multistage filter represent a naturally coarse level of granularity for investigating hardware reuse. The ability to hardwire and reuse these stages across multiple standards is limited by their dependency on standard specific parameters. The fixed factorization method proposed in this thesis factorizes any arbitrary integral or rational SRC factor into a regular form. This allows the band edge and stopband attenuation of all the filter stages prior to the last stage to be modified in a manner which allows them to be hardwired and reused across multiple standards with very low reconfiguration overheads. Experimental synthesis results have been used to demonstrate the area advantage of the proposed accelerator design over the Velcro approach and the power advantage over the coprocessor approach.

3. Low-area, low-power programmable time-shared filters

The last stage filter in the fixed factorization method needs to be fully reprogrammable to support standard dependent band edges, roll-off factors and stop band attenuations. Reprogrammability can be supported by time-multiplexing the filtering operations onto a set of generic MAC units. Such a filter can be reprogrammed by simply updating the coefficient memory. However time-shared filters with generic MAC units incur a significant dynamic power consumption

penalty when compared to a spatial style implementation of a constant coefficient filter. Hence the last stage filter dominates the overall power consumption of multistandard decimation filters designed using the fixed factorization method. We propose a class of time-shared filter derived from fast filter algorithm (FFA) structures [32, 33]. These filters exploit the property of algorithmic strength reduction offered by FFA structures to reduce the number of expensive MAC operations at the cost of increased add operations. The proposed class of time-shared filters are more area/power efficient than traditional direct-form based time-shared structures, when the levels of parallelism is increased.

1.4 Organization

This thesis is organized as follows. Chapter 2 gives an overview of the channelization function in the digital front-end in the context of a multistandard software defined radio handset. Chapter 3 introduces the notion of the flexibility penalty and analyzes the area, power and reconfiguration penalties of traditional multistandard filter accelerator design paradigms. In Chapter 4, we propose a novel channelization strategy that has a high degree of hardware reuse across multiple standards and low reconfiguration overheads. In Chapter 5, we propose a class of time-shared FIR filters that can be used for reducing the area/power penalty of the reprogrammable portions of the channelization accelerator. Chapter 6 gives our conclusions and directions for future work.

2

The Digital Front-end of Software Defined Radios

Achieving the vision of a flexible multistandard SDR handset presents significant problems at each point in the radio transceiver chain. The design of the DFE which acts as the interface between the analog front-end and the modem algorithms is particularly challenging. Firstly channel bandwidths have been increasing to support higher data rates. Secondly the bulk of the channelization load has been slowly shifting towards the digital side, especially with the rise of $\Sigma\Delta$ ADC. Both the trends have tremendously increased the computational load of the DFE. With emerging standards like LTE co-existing with legacy standards like GSM, a multistandard DFE has to be sufficiently flexible to support channels ranging from 200 KHz to 100 MHz. The major challenge facing SDR baseband designers is to incorporate such a high degree of flexibility, within the allowed power budget of mobile handset.

2.1 Analog Front-end Architectures

The design of digital front-end is strongly coupled to the design of the analog front-end and the ADC. Direct RF digitization imposes a prohibitively high dynamic range, linearity, sampling rate and full scale range (FSR) requirement on the ADC. Hence the analog front-end has the role of performing the tasks of signal amplification, coarse channel selection, frequency downconversion, anti-aliasing and image rejection. This

2.1 Analog Front-end Architectures

section gives an overview of the various analog front-end architectures from the perspective of the integrability and low power requirements of a multistandard radio.

2.1.1 Superheterodyne Receiver

The superheterodyne architecture is the traditional wireless receiver architecture that is commonly used for single mode radios [37, 38]. It is characterized by high levels of sensitivity and selectivity. It uses two or more frequency downconversion stages to bring the channel of interest to baseband. The basic architecture has been shown in Fig. 2.1. The acronyms BPF, LPF, LO and LNA in Fig. 2.1 stand for bandpass filter, low pass filter, local oscillator and low noise amplifier respectively. The superheterodyne architecture uses passive bandpass filter for removing the out of band interferers before signal amplification. The channel of interest is now downconverted to an intermediate frequency (IF) using a tunable local oscillator. Since real mixing suffers from the problem of superimposition of the image frequency channel with the required channel after downconversion, a high Q, off-chip image rejection filter has to be used to remove the image channel prior to mixing. A second high Q, off-chip surface acoustic wave (SAW) channel select filter is used to select the channel of interest at the IF, and is then shifted to baseband using quadrature conversion. This relaxes the specifications of the final baseband channel select filter. The main disadvantages of the above structure are the large number of discrete off chip components and very high power consumption. The low monolithic integration capabilities and the need for multiple off-chip filters for different standards make the super heterodyne architecture a poor candidate for a multistandard radio.

Figure 2.1: Superheterodyne receiver.

2.1 Analog Front-end Architectures

2.1.2 Zero-IF Receiver

The zero-IF receiver reduces the need for off chip discrete filters, by translating the channel of interest directly to baseband through a single quadrature downconversion stage as shown in Fig. 2.2 [39]. It solves the problem of superimposition of the image channel by signal cancellation, rather than the use of high Q, image reject filters. The image rejection requirements are more relaxed as the image frequency is the same as the desired channel. A tunable local oscillator is used to select the desired channel and a low pass channel select filter in each of the I and Q branches are used to remove the interferers and blockers. The zero-IF receiver has some well known disadvantages that had limited its widespread adoption, as compared to the super heterodyne receiver. Since the desired channel is directly translated to baseband, it is easily affected by low frequency interference like flicker noise, non linearities and IQ mismatches. Because the LO frequency and the carrier frequency of the desired channel are the same, it suffers from the problem of DC offset. A number of approaches that have been proposed to tackle these problems can be found in [40, 41, 42]. Since it needs no off-chip components, the zero-IF architecture offers a high level of monolithic integration and low power consumption, making it a good candidate for multistandard radios.

Figure 2.2: Zero-IF receiver.

2.1.3 Low-IF Receiver

The low-IF receiver too offers a high level of integration, but overcomes the problems associated with low frequency interference by using a single tunable quadrature

2.1 Analog Front-end Architectures

downconversion stage to shift the channel of interest to a low intermediate frequency [37, 43, 44], as shown in Fig. 2.3. The low-IF is typically chosen to be one or two channels away from DC. Since the image channel has to be removed by signal cancellation, the low-IF architecture has more stringent image rejection requirements than the zero-IF receiver and hence is more susceptible to IQ mismatches. The digitized channel is effectively a bandpass signal, thus requiring double the bandwidth to be processed by the ADC when compared to a zero-IF receiver. Consequently, the digitization incurs a higher power consumption penalty when compared to the zero-IF receiver. The low-IF receiver is also a good candidate for multistandard radios, especially for narrow channel bandwidths.

Figure 2.3: Low-IF receiver.

2.1.4 Wideband-IF Receiver

Figure 2.4: Wideband-IF receiver.

2.2 Digitization in a Multistandard Radio

The wideband-IF receiver also uses two analog downconversion stages like the classical super heterodyne receiver [45]. The main difference is that, instead of a tunable LO in the first downconversion stage, a fixed LO is used, thereby shifting the entire band selected by the RF bandpass filter to the IF. A low pass filter is used to remove the high frequency components. A tunable LO is then used to select the desired channel and translated to DC using a quadrature downconversion stage, as shown in Fig. 2.4. Only a fixed LO is used in the first stage while the tunable LO operates at a lower frequency which permits, more phase noise reduction at the LO outputs [46]. Potential self mixing in the two mixer stages result in constant DC offsets which can be easily removed in the analog or digital domain. The architecture allows a narrow channel to be selected without high Q off-chip SAW channel select filters, hence offering higher levels of integration. Part of channelization can also be shifted to the digital domain, which makes it a viable front-end alternative for multistandard radios.

2.2 Digitization in a Multistandard Radio

In all the analog front-ends discussed above, the RF signal goes through one or more analog filtering stages prior to the analog-to-digital conversion. If channel selectivity is completely provided in the analog domain, the channel selection filters would need to have a high degree of selectivity in addition to being programmable over a wide range of channel bandwidths to accommodate multiple standards. The need for such programmable analog filters can become a potential bottleneck for infield upgradability. A more efficient analog front-end design paradigm for multistandard radios would be to use a ‘fixed digitization bandwidth’ [20]. Under this methodology, instead of reprogramming the channel selection filters for every possible channel bandwidths, the bandwidth is fixed according to an estimate of the maximum possible channel bandwidth. This considerably relaxes the flexibility requirement of the analog filters, while shifting the fine channel selectivity to the digital domain, where it is easier to provide reprogrammability. However, the input to the ADC can now be several channel bandwidths wide, especially for narrow band signals, as shown in Fig. 2.5. The adjacent channel interferers and blockers can be much more powerful than the desired signal, imposing stringent dynamic range and sampling rate requirements on the ADC. The

2.2 Digitization in a Multistandard Radio

Figure 2.5: ADC input in the fixed digitization bandwidth approach for different standards.

interferer and blocker frequency masking profile for the GSM standard are shown in Fig. 2.6 and Fig. 2.7 respectively, to illustrate the above point.

Figure 2.6: GSM Interferers.

The dynamic range of Nyquist ADCs which sample at close to twice the input bandwidth can be increased either by increasing the bit resolution or the sampling rate. Both the above operations result in a significant increase of power consumption, which depends quadratically on the accuracy and linearly on the speed [29]. The bit resolution is largely determined by the matching of the analog components. Increasing the resolution by a single bit results in a fourfold increase in power consumption. The Nyquist ADC is also inefficient and wasteful for digitizing a wideband signal containing

2.2 Digitization in a Multistandard Radio

Figure 2.7: GSM Blockers.

both the wanted signal and unwanted interferers, as it digitizes the entire band with a high dynamic range as shown in Fig. 2.8, while it may be required only in a small frequency band for standards with narrow channel bandwidths.

Figure 2.8: Distribution of Quantization Noise in Nyquist ADCs.

$\Sigma\Delta$ ADCs are more efficient in scenarios like the one above due their ability to provide a very high dynamic range only within the channel of interest [21, 47]. They use only a low bit quantizer, which normally results in large amount of quantization noise. In a $\Sigma\Delta$ ADC, the quantization noise is spread over a very wide band by oversampling the desired channel at much higher than its Nyquist rate. The SNR improves only by 3dB for each doubling of the oversampling ratio (OSR). Further improvement in SNR is achieved by employing a negative feedback loop, to shape the quantization noise away from the desired channel to higher frequencies, where the unwanted interferers and blockers lie, as indicated in Fig. 2.9.

Increasing the order of noise shaping results in further reduction of the in-band quantization noise [48]. The $\Sigma\Delta$ ADC by itself cannot provide a high dynamic range, as it only shapes the quantization noise, without removing it. The task of removing the quantization noise is performed by a subsequent sharp digital filter, resulting in a high dynamic range in the channel of interest. For an arbitrary L th order $\Sigma\Delta$ ADC, the

2.2 Digitization in a Multistandard Radio

Figure 2.9: Distribution of Quantization Noise in $\Sigma\Delta$ ADC.

dynamic range can be given as follows [48]:

$$DR = \frac{3}{2} \left(\frac{2L+1}{\pi^{2L}} \right) G^2 M^{2L+1} (2^B - 1)^2 \quad (2.1)$$

In Eq. 2.1 the parameter G represents the gain in the forward path of the $\Sigma\Delta$ ADC, M represents the OSR and B represents the quantizer bit resolution. Doubling the OSR, results in an increase in the bit resolution by $L + 0.5$ bits.

For high channel bandwidths, the required oversampling rates might be unacceptably high for practical implementation. For such cases topologies like the multibit and cascaded $\Sigma\Delta$ ADCs have been reported in literature, which can provide a high dynamic range even with relatively low OSRs [49, 50]. The $\Sigma\Delta$ ADC is inherently reconfigurable, providing a tradeoff between bandwidth and dynamic range. Various multistandard $\Sigma\Delta$ ADCs have been proposed whose parameters like OSR, noise shaping order, number of cascade stages and quantizer bit resolution can be reconfigured to provide multistandard support [30, 31]. A class of $\Sigma\Delta$ ADC known as continuous time sigma delta (CT- $\Sigma\Delta$) converters, use continuous time filters rather than discrete time filters to perform the noise shaping [51], as done in traditional $\Sigma\Delta$ ADCs. These converters perform an implicit anti-alias function, thereby reducing the load or even eliminating the need for an analog anti-aliasing filter. This can lead to highly integrated multistandard radios. The traditional $\Sigma\Delta$ ADC presents a low pass signal transfer function and a high pass noise transfer function. The same idea can be extended to cases where the channel of interest is not at DC, like the low-IF receiver. The bandpass sigma delta ADC [52] can be used for such cases, which presents a bandpass signal transfer function for the desired signal, while presenting a bandstop response for the quantization noise.

2.3 The Digital Front-end Functions

The “digital front-end” is a term coined in [20], to represent the interface between the ADC and the subsequent digital baseband blocks. It essentially performs the task of channelization, which includes all the signal processing tasks that are required to extract a weak channel of interest from a highly oversampled signal in the presence of large interferers, blockers and out-of-band quantization noise. The actual tasks involved are outlined below.

- Perform a complex digital mixing operation to downconvert desired channel to DC.
- Select a weak channel of interest from a highly oversampled signal in the presence of strong interferers.
- Perform a sample rate conversion operation to decimate the oversampled input from the ADC to the standard specific symbol/chip rate of the current standard. SRC factors can be integral or rational.
- Ensure that the in-band noise due to aliasing in the sample rate conversion operation is below maximum co-channel interference level for each standard.
- Ensure that all interferers, blockers and quantization noise are attenuated below the carrier to noise (C/N) ratio for each standard.

The SDR is an enabling technology for communication paradigms like cognitive radios, which require the underlying radio functional blocks to be extremely flexible. The cognitive radio terminal has to sense the spectrum over multiple bands having channels of varying bandwidth, as shown in Fig. 2.10, which illustrates the sensing operation over two different bands. When multiple vacant channels are available, it is desirable to select the widest channel or fuse multiple contiguous narrow channels to obtain a wider channel bandwidth. A wider vacant channel offers more protection against flat fading and greater opportunities for frequency diversity schemes which can offer protection against frequency selective fading [14]. Since the vacant channel used during a transmission interval can vary over time, the worst case power levels of the adjacent channel interferers around the vacant channel can also vary with time. Hence the

2.3 The Digital Front-end Functions

digital front-end has to be flexible enough to accommodate varying channel bandwidths, passband and stopband edges, adjacent channel interference levels, C/N ratios and SRC factors.

Figure 2.10: Spectrum sensing over two bands with different channel bandwidths.

2.3.1 Digital Downconversion

In zero-IF or wideband-IF receivers, the tunable LO of the analog quadrature down-conversion stage can be chosen to directly translate the desired channel into DC. Hence low pass filtering can be used to remove the interferers and quantization noise, and the signal can be decimated to the symbol rate/ chip rate of the current mode of operation. For low-IF receivers, the desired channel is digitized at a slight frequency offset from DC, hence an additional quadrature digital downconversion stage is required to first translate the desired channel to DC, before the filtering and sample rate conversion operations.

Assume that the desired channel is at the frequency f_{IF} . Let $\omega_{IF} = 2\pi \left(\frac{f_{IF}}{f_S} \right)$, where f_S is the sampling rate. The input signal, $X_I + jX_Q$ is a complex digital signal, with the real and imaginary parts representing the digitized samples from the I and Q branches. The desired channel can be shifted to baseband by multiplying the input with the complex exponential, $e^{-j\omega_{IF}n} = \cos(\omega_{IF}n) - j\sin(\omega_{IF}n)$. The output, $Y_I + jY_Q$, of the complex mixing operation can be represented by Eq. 2.2 and Eq. 2.3 [53].

$$Y_I = X_I \cos(\omega_{IF}n) + X_Q \sin(\omega_{IF}n) \quad (2.2)$$

$$Y_Q = X_Q \cos(\omega_{IF}n) - X_I \sin(\omega_{IF}n) \quad (2.3)$$

2.3 The Digital Front-end Functions

The required complex digital mixing operation requires four real multipliers to be implemented in practice as shown in Fig. 2.11. The sinusoidal values are generated by a numerically controlled oscillator (NCO) which produces the sampled values of a sinusoid having a precise frequency and phase. The required sinusoid values are obtained by storing a phase to amplitude mapping in a ROM based lookup table. For high

Figure 2.11: Digital downconversion of a complex input signal.

resolution, the required ROM tables occupy a large chip area and have high power consumption [53]. There is additional power consumption overhead due to the four multipliers operating at the input oversampled rate. Both the above overheads can be avoided by a clever choice of the intermediate frequency and input sampling rate. When f_S and f_{IF} are chosen such that $\frac{f_{IF}}{f_S} = 1/4$, ω_{IF} reduces to $\frac{\pi}{2}$. The corresponding sampled sinusoids can be represented by periodic sequences comprising of 1, 0 or -1. This eliminates the need for general multipliers or large ROM tables for the NCO.

2.3.2 Pulse Shaping

Transmitting information over a band-limited wireless channel introduces bit-errors due to inter symbol interference (ISI), which needs to be taken into account by the channelization function. Band-limiting a rectangular pulse introduces ripples in the time domain, which can interfere with the symbol estimation operation. Combating ISI requires the use of pulse shaping filters like raised cosine (RC) filters or Gaussian filters [54]. These filters shape the frequency response of the transmitted rectangular pulse such that the zero crossings of the time domain ripples coincide with the midpoint of the neighboring symbols, thereby causing no inter symbol interference (ISI), if the symbol decision is taken at the midpoint. In addition to this, the pulse shaping filters

2.3 The Digital Front-end Functions

also reduce ISI due to multipath reflections, by attenuating the start and end portions of the symbol, which are most likely to overlap with the midpoints of the neighboring symbols [54]. The pulse shaping filter determines the effective passband within which the useful information lies. The SRC operation performed by the DFE should ensure that this band is protected from aliasing. The frequency response of a RC filter is governed by its roll-off factor. The roll-off factor, α lies in the range $0 \leq \alpha \leq 1$, and provides a tradeoff between the frequency bandwidth and the magnitude of the time domain ripples. The passband edge, f_{pass} and stopband edge, f_{stop} of the RC filter can be expressed in terms of the pulse rate, f_0 , (which is equal to the chip rate in spread spectrum systems and the symbol rate in non-spread spectrum systems) and the roll off factor as follows.

$$f_{pass} = \frac{f_0}{2} (1 - \alpha) \quad (2.4)$$

$$f_{stop} = \frac{f_0}{2} (1 + \alpha) \quad (2.5)$$

Gaussian pulse shaping filters do not have an explicit roll-off factor. They are typically specified in terms their bandwidth-time (BT) product, which is defined as the product of the 3dB bandwidth and the symbol period. Reducing the BT product has the effect of reducing the occupied bandwidth at the cost of increased ISI. Gaussian filters are typically used in scenarios where some amount of ISI is tolerable, and for most of the commonly used standards, the BT product lies in the interval, $[0,0.5]$ [135]. The corresponding band edge of the information bandwidth that needs to be protected from aliasing can be given by given in terms of f_0 and the BT product, β .

$$f_{pass} = f_0 \beta \quad (2.6)$$

When a RC response is used for pulse shaping , the total pulse shaping load is usually split between matched transmit and receive square root raised cosine (SRRC) filters. When implemented in the digital domain, the SRRC filter response is usually achieved through a finite impulse response (FIR) filter, mainly due to its stability, the ease of achieving linear phase characteristics and the absence of limit cycle issues. The maximum possible bandwidth in a SRRC filter occurs for $\alpha = 1$, where the stopband edge is given by $f_{stop} = f_0$. To satisfy Nyquist criterion in a digital system, the

2.3 The Digital Front-end Functions

minimum oversampling rate at the input of the SRRC filter should be two [54]. The cost of implementing a FIR filter can be measured in terms of the number of filter taps. Since the roll-off factor can vary across different standards, the pulse shaping filter needs to be reprogrammable. This requires the coefficient multiplications corresponding to each filter tap to be performed using a generic multiplier. An N tap FIR filter needs to perform N multiplications and $N - 1$ additions per output sample. The length of an FIR filter is inversely proportional to its normalized transition bandwidth [55]. For an input sample rate of $f_S = Mf_0$, the normalized transition band of a SRRC filter can be given in terms of its roll-off factor as:

$$f_{transition} = \frac{f_0}{f_S} \alpha = \frac{\alpha}{M} \quad (2.7)$$

where M stands for the OSR. When $\Sigma\Delta$ ADCs are used for performing the analog-to-digital conversion, depending on the relation between the ADC sampling rate and the symbol rate, M can be expected to be very high integral or rational factor. This results in a very narrow transition band and consequently a very high order filter. This filter in turn has to operate on the ADC output arriving at the oversampled rate. This results in very high power consumption due to the large number of computations. However a significant number of computations go waste, as the output has to be decimated to the symbol rate/chip rate by before feeding it to the subsequent modem and codec baseband processing blocks. Assume that the SRRC channel filter has been designed for an oversampling rate of M_S . A more efficient configuration than the one discussed above, would be to first perform the sample rate conversion operation to decimate the ADC sampling rate to $M_S f_0$ and then perform the channel filtering using the pulse shaping filter. The SRRC stage performs the final decimation by a factor of M_S , as shown in Fig. 2.12. This ensures that SRRC channel filtering is performed in a relatively low OSR environment, while sample rate converter (SRC) operates in a high OSR environment. The pulse shaping parameters of some of the commonly used

Figure 2.12: Pulse shaping in the receiver chain.

2.3 The Digital Front-end Functions

standards is given in Table 2.1.

Table 2.1: Pulse shaping parameters

Standard	Pulse shaping	BT/ α	Receive-side pulse shaping filter
GSM	Gaussian	BT = 0.3	No
W-CDMA	Raised Cosine	$\alpha = 0.22$	Yes
IEEE 802.11a	Gaussian	BT = 0.5	No
WiMAX	Raised Cosine	$\alpha = 0.25$	Yes

2.3.3 Sample Rate Conversion

The ADC sampling clock is typically independent of the symbol rate/chip rate of the standard. In general when oversampling based $\Sigma\Delta$ ADCs are used for digitization, the digital front-end has to perform sample rate conversions by large rational factors. A textbook approach to the problem of sample rate conversion by an arbitrary rational factor would be to first represent it as a rational number $\frac{P}{Q}$, where P and Q are integers. Decimation by the factor $\frac{P}{Q}$ can be achieved by first upsampling the input signal by a factor of Q , followed by a downsampling by a factor of P [56]. The interpolation and decimation filters required for the upsampling and downsampling operations respectively can be combined into a single filter. The main drawback with this simple approach stems from the fact that P and Q are typically very large integers. Hence the required intermediate frequency after the first upsampling operation might be too impractical to implement it on real hardware.

In general any SRC system, whether integral or rational can be modeled as a reconstruction of the discrete time samples by a continuous time reconstruction filter followed by a resampling of the continuous time waveform at the required output rate. Assume that the input sample period and the output sample period are T_{in} and T_{out} respectively. Mathematically the operation can be shown as follows. If $h(t)$ represents the reconstruction filter, and $y(nT_{in})$ represent the discrete input samples, the reconstruction operation can be represented by the following relation [60]:

$$y(t) = \sum_{n=-\infty}^{\infty} y(nT_{in}) h(t - nT_{in}) \quad (2.8)$$

2.3 The Digital Front-end Functions

Resampling the above continuous waveform, $y(t)$, at the instant kT_{out} achieves the required sample rate conversion as shown below:

$$y(kT_{out}) = \sum_{n=-\infty}^{\infty} y(nT_{in}) h(kT_{out} - nT_{in}) \quad (2.9)$$

The output sample instants can be expressed in terms of the nearest input sample as follows. Let n_k be the largest integer such that $n_k T_{in} \leq kT_{out}$. Then, n_k can be given by Eq. 2.10

$$n_k = \left\lfloor \frac{kT_{out}}{T_{in}} \right\rfloor \quad (2.10)$$

A quantity called the fractional inter sample interval μ_k can be defined as shown in Eq. 2.11 and Eq. 2.12. It represents the interval, that exist between the output sample instant and the nearest input sample event, as shown in Fig. 2.13.

$$kT_{out} = \mu_k T_{in} + n_k T_{in} \quad (2.11)$$

$$\mu_k = \frac{kT_{out}}{T_{in}} - n_k \quad (2.12)$$

Figure 2.13: Input and output sample instants for arbitrary sample rate conversion.

A practically realizable reconstruction filter needs to have a finite time domain response. Assume that the filter response is non zero only in the interval $(-N_1 T_{in}, N_2 T_{in})$. Eq. 2.8 can be recast as follows:

$$y(kT_{out}) = \sum_{l=-N_1}^{N_2} y((n_k - l) T_{in}) h((\mu_k + l) T_{in}) \quad (2.13)$$

2.3 The Digital Front-end Functions

The reconstruction filter in Eq. 2.13 has a time-varying impulse response, which depends on the fractional sample interval μ_k . Reconstructing the entire input band, using a time-varying filter, can be an extremely inefficient way of performing the sample rate conversion, as the desired channel may only be small fraction of the total digitized band. The theoretical ‘interpolation model’ of sample rate conversion by a rational factor is also not able to handle the large rational decimation factors typically encountered in the digital front-end. While an ideal reconstruction filter is able to completely remove all images in the input caused by the analog-to-digital conversion process, a practically realizable reconstruction filter only needs to provide image attenuation in a small band around the image frequencies $\frac{n}{T_{in}}$, $n \in \{1, 2, 3, \dots\}$. When the above system is used for decimation by a large rational factor, the unattenuated quantization noise and interfering channels which may lie around the frequencies $\frac{n}{T_{out}}$, $n \in \{1, 2, 3, \dots\}$, can alias into the desired channel.

Practical SRCs based on the above model, like the Farrow Structure are more useful for handling fractional SRC factors which are very close to 1 [60]. Consider the case of an integral decimation, from an input sample rate f_{in} to an output sample rate f_{out} . Let $f_{in} = M f_{out}$. Under these conditions, Eq. 2.10 and 2.13 can be reduced to the expressions Eq. 2.14 and 2.15 respectively.

$$n_k = \frac{kT_{out}}{T_{in}} = \frac{kMT_{in}}{T_{in}} = kM \quad (2.14)$$

$$y(kT_{out}) = \sum_{l=-N_1}^{N_2} y((kM - l)T_{in}) h(lT_{in}) \quad (2.15)$$

It can be seen from Eq. 2.14 and 2.15 that a time-varying reconstruction filter is no longer required when the decimation factor is an integer. Integral decimation is a well studied problem, having many efficient solutions available in the literature [61]. The filtering structures typically used for integral decimation offer better control over the attenuation of the potential aliasing bands. Hence it is important to extract the integral and fractional SRC conversion loads from the original rational decimation factor, and perform the conversions separately.

2.3 The Digital Front-end Functions

2.3.4 Interferer Attenuation

The SRC and pulse shaping stages implicitly perform a channel selection and interferer attenuation function. Shifting the fine-channel selection load to the digital domain allows the specifications of the analog front-end to be relaxed. In such a situation the output of $\Sigma\Delta$ ADC may comprise of several unwanted interferer channels and a high level of out of band quantization noise in addition to the channel of interest. Preserving the fidelity of the desired channel for successful demodulation requires the filtering and sample rate conversion operations to be performed while maintaining a standard specific carrier to noise ratio ($\frac{C}{N}$) [62]. All potential interferers and blockers that have not been filtered by the analog front-end, as well as the additional quantization noise introduced by the $\Sigma\Delta$ ADC, have to be sufficiently attenuated to satisfy the $\frac{C}{N}$ constraints. The $\frac{C}{N}$ is a function of the bandwidth efficiency and the power efficiency of the modulation scheme used by the current standard, and is commonly given by the following expression [34].

$$\frac{C}{N} = \left(\frac{E_b}{N_0} \right) \left(\frac{R}{W} \right) \quad (2.16)$$

In Eq. 2.16, E_b represents the energy per bit, N_0 represents the noise power spectral density, R represents the bit rate, and W represents the bandwidth. The $\frac{E_b}{N_0}$ ratio needed to satisfy a specific bit error rate (BER) requirement, represents the power efficiency of the modulation scheme, while $\frac{R}{W}$ ratio indicates its bandwidth efficiency. The power levels of the potential interferers and blockers can be several times the power level of the desired signals. In a sample rate conversion system, interferer/noise attenuation is of added importance, as the insufficiently attenuated unwanted signals could potentially alias back into the desired channel, irrecoverably corrupting it. Hence the most important constraint in the digital front-end is to prevent aliasing into the channel of interest [64]. Providing multistandard support in a multimode channelization accelerator is complicated by the fact that the position of the interferers, their power levels, the channel bandwidth that has to be protected from aliasing, as well as the $\frac{C}{N}$ requirements can show considerable variation.

2.4 Specialized Filter Structures For Channelization

Once the desired channel has been downconverted to DC, the primary channelization tasks consist of channel selection, interference and alias attenuation, sample rate conversion and pulse shaping, which can all be reduced to a set of FIR filtering and decimation operations. The length of a FIR filter, N_l , is inversely proportional to its normalized transition bandwidth, and can be given by Eq. 2.17 [55].

$$N_l = \frac{2}{3} \cdot \log \left(\frac{1}{10\delta_1\delta_2} \right) \cdot \frac{f_s}{\Delta f}, \quad (2.17)$$

where δ_1 is the passband ripple, δ_2 is the stopband ripple, f_s is the sampling rate and Δf is the transition bandwidth. $\Delta f/f_s$ represents the normalized transition bandwidth. Since $\Sigma\Delta$ ADC sampling rate, f_s , is typically very high, a single stage decimation filter has a very narrow normalized transition bandwidth, resulting in a high order filter. Decimating in multiple stages ensures that only lower order filters operate in a high OSR environment while higher order filters need to operate only on a low OSR signal, reducing the overall computational complexity. Splitting the decimation load into multiple stages also allows one to exploit certain specialized low complex filter structures which are useful only for particular decimation factors or in specific positions in the decimation chain.

2.4.1 Cascaded Integrated Comb (CIC) Filter

In conventional multistage decimation filter designs, the CIC filter has usually been used as the first stage. The transfer function of the CIC filter can be given as follows [65].

$$H(z) = \left(\frac{(1 - z^{-R_d})}{R_d(1 - z^{-1})} \right)^{N_c} \quad (2.18)$$

In Eq. 2.18, R_d represents the decimation factor of the CIC filter. The order of the CIC filter N_c , represents the number of cascaded 1st order CIC stages. The frequency response corresponding to the above transfer function can be given as follows.

$$H(\omega) = \left(\frac{\sin(\omega R_d/2)}{R_d \sin(\omega/2)} e^{-j\omega(R_d-1)/2} \right)^{N_c} \quad (2.19)$$

2.4 Specialized Filter Structures For Channelization

The above transfer function has zeros in all potential aliasing bands, which are clustered around integral multiples of $\frac{2\pi}{R_d}$. The classical IIR-FIR implementation of the CIC filter as shown in Fig. 2.14, consist of a set of cascaded integrator stages followed by as set of comb filter stages with the decimator in between [65].

Figure 2.14: Cascaded Integrator Comb Structure.

The main attraction of the CIC filter as a first stage is that it has unity-valued coefficients and hence has no multipliers operating at the oversampled rate. The structure is also inherently programmable as any change in the decimation rate can be achieved by just changing the sampling clock of the decimator between the integrator and comb filters without any change in hardware. Changing the order of the CIC filter can also be easily achieved by powering down some of the integrator and comb stages. The main disadvantage of the classical recursive CIC implementation is need for large wordlength adders operating at the oversampled rate which results in high power consumption. Due to bit growth in the integrators for correct operation, the wordlength W_l of the adders and the registers has is given by $W_l = w_0 + N_c \log_2 R$, where w_0 is the input wordlength [66]. Since the required decimation factor R_d is typically very large for sigma delta based converters the resulting wordlength requirement is also large.

For the cases where R_d is a power of 2, very efficient non recursive implementations have been proposed [66]. The non recursive implementation can be obtained by factorizing the CIC transfer function as shown below.

$$\left[\frac{(1 - z^{-R_d})}{R_d (1 - z^{-1})} \right]^{N_c} = \left[\frac{1 + z^{-1} + z^{-2} + \dots + z^{-R_d+1}}{R_d} \right]^{N_c} \quad (2.20)$$

When $R_d = 2^s$, the RHS of Eq. 2.20 can be factorized as shown below.

$$H(z) = \left[\frac{1}{2} (1 + z^{-1}) \right]^{N_c} \left[\frac{1}{2} (1 + z^{-2}) \right]^{N_c} \left[\frac{1}{2} (1 + z^{-4}) \right]^{N_c} \left[\frac{1}{2} (1 + z^{-2^{s-1}}) \right]^{N_c} \quad (2.21)$$

The succeeding decimation by 2^s can be shifted to the left using Noble identity [56], resulting in the structure shown in Fig. 2.15. There are totally s stages where $R_d = 2^s$.

2.4 Specialized Filter Structures For Channelization

The advantage of this structure is that the bit growth in each stage is only N_c bits. Notice that while the wordlength requirement of each stage progressively increases, the operating rate also drops by 2 for each stage. This results in a significant reduction in power consumption over the recursive CIC implementation.

Figure 2.15: Nonrecursive CIC structure.

The term $\left[\frac{1+z^{-1}}{2} \right]^{N_c}$ can be expanded and implemented in polyphase form, which further reduces the operating rate of the filter by two [67]. Since the filter coefficients are typically small integers, they can be implemented using simple constant integer multipliers, as shown in Fig. 2.16, for a fifth order CIC filter (The ' $\ll X$ ' and ' $\gg X$ ' symbols represent a left shift by X and a right shift by X respectively). Since the number of stages, s , in the NRCIC structure is determined by the decimation factor 2^s , variable decimation factors can be supported by bypassing one or more stages.

Figure 2.16: Polyphase implementation of the term $\left[\frac{1+z^{-1}}{2} \right]^5$

It is difficult to obtain a general expression for the total number of adders in the

2.4 Specialized Filter Structures For Channelization

polyphase subfilters of $\left[\frac{1+z^{-1}}{2}\right]^{N_c}$, since it is a function of both the filter coefficients and opportunities for hardware sharing between the coefficients. The complexity of the fifth order NRCIC filter is analyzed below. Each of the s stages contains 7 adders, as shown in Fig. 2.16. Each stage operates at half the sample rate of the previous stage. Assuming that the input sample rate is F_s . The total number of additions per second can be given by:

$$\sum_{p=1}^s 7F_s/2^p = 7F_s \left(1 - \frac{1}{2^s}\right) = 7F_s \left(1 - \frac{1}{R_d}\right) \quad (2.22)$$

With the above transformations, the CIC filter offers a cost effective alternative to a conventional FIR filter, for the first stage decimation in a multistage decimation filter. But it also has two predominant drawbacks that have to be taken into account while using it in a practical design. CIC filters do not have a flat passband. The sinc shaped passband droop depends upon the required decimation factor and the CIC filter order as can be observed from Eq. 2.19. They also have a non uniform attenuation in the stopband. The attenuation is maximum in the centre of the aliasing bands where the zeros of the CIC cascade are clustered and considerably lower at the band edges as shown in Fig. 4.8.

Figure 2.17: CIC filter magnitude response.

If a simple FIR filter is present in the multistage decimation filter chain, the first problem of a passband droop can be easily corrected by shaping the passband of the FIR stage to have an inverse sinc frequency characteristic [69]. This incurs no additional penalty. Since the edge of the passband that has to be corrected and the CIC order can vary across standards, the droop correction filter needs to be reprogrammable. If

2.4 Specialized Filter Structures For Channelization

the CIC filter is a part of a decimation chain where the subsequent stages do not offer any control over the passband, the above strategy cannot be used. In [68], a structure called the sharpened CIC filter was introduced which has a lower passband droop and improved attenuation in the aliasing bands. The main drawback of the approach is the significant increase in complexity. A sharpened version of a N_c th order CIC filter, has nearly the three times the complexity of a single N_c th order CIC [68]. Because of the cascade nature of the sharpened CIC structure [68], the non recursive polyphase transformations cannot be readily applied to reduce the power consumption. To tackle the problem of non-uniform stopband attenuation, a structure called the rotated sinc filter was proposed in [70]. The above structure modifies the CIC filter response by re-distributing the zeros over the entire aliasing band resulting in significant improvement of SNR. The disadvantage of this structure is the need for multipliers operating at the oversampled rate, thus negating the main attraction of using a CIC filter in the first stage. An alternative strategy to modifying the CIC filter response has been proposed in [71], where a low cost post-filter is used to improve the CIC characteristics. The approach uses a class of second order polynomials which have monotonically increasing characteristics in the CIC passband. Hence a cascade of the interpolated second order polynomial (ISOP) filters and the CIC filters has a reduced passband droop, at the expense of a slight deterioration of the alias attenuation. The ISOP filter is a 3-tap filter, with two unity coefficients. Hence it has an overhead of only one coefficient multiplier operating at the decimated rate after the CIC filter. The ISOP stage needs to be reprogrammable for supporting varying CIC orders or decimation factors. In [72], the authors propose a multiplierless wideband CIC compensation filters which is very similar to the ISOP filter. The authors of [72] also use a symmetric 3-tap FIR filter, which compensates the CIC response within the interval $(-\pi/M, \pi/M)$. The filter coefficients are obtained by an iterative algorithm. The multiplierless implementation is obtained by rounding the coefficients with a rounding factor, and implementing the resultant integer coefficients using constant integer multipliers. While the structure is efficient for a fixed decimation factor and a fixed CIC order, the dependency of the filter coefficients on both the CIC decimation factor and the CIC order precludes a multiplierless implementation when these parameters are variable.

2.4 Specialized Filter Structures For Channelization

2.4.2 Farrow Structure

The Farrow structure [73] has been commonly used for fractional sample rate conversion and is based on the interpolation model of sample rate conversion represented by Eq. 2.13, which is reproduced below.

$$y(kT_{out}) = \sum_{l=-N/2}^{N/2-1} y((n_k - l)T_{in}) h((\mu_k + l)T_{in}) \quad (2.23)$$

To implement the time-varying impulse response required in the above equation, the underlying continuous time reconstruction filter response, which is assumed to be non zero only in the interval $(-(N/2)T_{in}, (N/2 - 1)T_{in})$, can be modeled in each T_{in} interval using a piecewise polynomial in μ_k . The advantage of such an approximation is that such polynomial filters can be efficiently be implemented using the Farrow structure, shown in Fig. 2.18 [60, 73]. The advantage of the Farrow structure is that the requirement of a time-varying interpolation filter, is replaced by a structure which consists of low order linear time invariant (LTI) subfilters. The only time-varying components are the μ -multipliers which have to be updated in every input clock cycle.

Figure 2.18: Farrow structure.

The computational complexity of an Farrow structure is analyzed below. Assuming the order of the underlying polynomial is M , the structure has $M + 1$ LTI FIR subfilters, each of length N (Note that NT_{in} is the time interval over which the reconstruction filter's time domain response is non-zero). Each FIR subfilter has N coefficient multipliers and $N - 1$ adders. In addition to the LTI subfilters, the structure also has M time-varying μ -multipliers. The subfilters operate at the input sample rate, while the μ -multipliers which operate at the output rate. Assume that the input sample rate is F_s and fractional SRC factor is R_f . The total number of multiplications per second

2.4 Specialized Filter Structures For Channelization

can be given by $F_s((M+1)N + MR_f)$. The total number of additions per second can be given by $F_s(M+1)(N-1)$.

When the length of the underlying impulse response is an integral multiple of T_{in} , the zeroes of the Farrow structure lie at the integral multiples of $\frac{1}{T_{in}}$. So the Farrow structure has good anti imaging properties. However, it performs poorly for fractional decimation, due to insufficient attenuation at the potential alias frequencies. In [136], the authors noted that by using an underlying impulse response whose length is an integral multiple of T_{out} rather than T_{in} , the zeros of the Farrow structure are clustered around multiples of $\frac{1}{T_{out}}$, resulting in good anti-aliasing properties. The coefficients of this structure can be obtained by a simple transposition of conventional Farrow structure coefficients and hence it has been referred to as the Transpose Farrow structure. To design the underlying polynomial filters, either classical time domain interpolating polynomials [74] or frequency domain design methods [75] can be used. The Farrow coefficients for time domain interpolating polynomials like Lagrange or cubic B-spline are fixed and hence can be reused for multiple fractional sample rate conversion factors. Switching between standards only requires the counters generating the fractional interval value to be reinitialized. However these polynomials have a poor response for wideband signals, which is the case when the Farrow structure is operating in a low OSR environment. Frequency domain synthesis techniques [75], optimize the Farrow structure coefficients, such that the underlying polynomial approximates the ideal reconstruction filter using a least square error or minimax error criterion. Frequency domain methods also allow an explicit passband and stopband edge to be defined which is not possible with the time domain methods. However the filter coefficients being different for different set of specifications, a hardware architecture that can support variable lengths and programmable coefficients would be required in a multistandard receiver.

2.4.3 Halfband Filter

Halfband filters are very efficient structures for integral decimation by a factor of two. Linear phase FIR halfband filters are constrained to be of odd length and have a transition band which is symmetrical around 0.5π radians. Hence they can be used only in situation which can tolerate some aliasing in the transition band. An equiripple linear phase FIR halfband design has an equal passband and stopband ripple [56]. The

2.5 Implementation of the DFE in existing SDR baseband architectures

impulse response of the halfband filter of length N (N is constrained to be odd) has the following property which makes it attractive for hardware implementation [56].

$$h_b(n) = \begin{cases} 0 & \text{if } n - (N - 1)/2 \text{ is even} \\ 1/2 & \text{if } n = (N - 1)/2 \end{cases} \quad (2.24)$$

It can be seen from Eq. 2.24 that approximately half the number of coefficients are zero. Hence it has half the number of multiply operations when compared to a conventional linear phase FIR filter with comparable specifications. Further reductions in operating rate by a factor of two can be obtained by implementing it in polyphase form.

The computational complexity of a N -tap halfband filter after the polyphase decomposition can be evaluated below. Note one of the subfilters has only a single non-zero coefficient of 0.5, which can be implemented as a simple hardwired shift, and hence does not need any multiplier or adder. The other subfilter has $(N + 1)/2$ taps. The polyphase decomposition of an odd length linear phase halfband filter has linear phase symmetrical subfilters. Hence even the polyphase implementation can exploit the reduction in multiplications due to coefficient symmetry. If $(N + 1)/2$ is even, the subfilter requires require $(N + 1)/4$ multipliers. If $(N + 1)/2$ is odd, the subfilter requires require $(N + 3)/4$ multipliers [137]. $(N - 1)/2$ structural adders are used within the subfilter, while one adder is used to combine the results from the two subfilters in the polyphase structure. Assuming that the input sample rate is F_s , for even $(N + 1)/2$, the number of multiplications per second is given by $(N + 1)F_s/8$, while for odd $(N + 1)/2$, the number of multiplications per second is given by $(N + 3)F_s/8$. The total number of additions per second in both cases is $(N + 1)F_s/4$.

The main properties of the three specialized filter structures are summarized in Table 2.2. The computational complexity of the NRCIC filter is specified for a 5th order NRCIC filter.

2.5 Implementation of the DFE in existing SDR baseband architectures

In 2G wireless radios, the radio baseband largely followed an ‘application specific integrated circuit (ASIC) centered, DSP assisted’ model, where the signal processing

2.5 Implementation of the DFE in existing SDR baseband architectures

Table 2.2: Properties of the Specialized filter structures

Structure	Pros	Cons	Mult/s	Add/s
NRCIC	Power Efficient	- Limited to powers of 2 - Useful only at High OSR - Passband droop	-	$7F_s(1 - 1/R_d)$
Farrow's structure	Handles arbitrary fractional factors	Complex structure	$F_s((M + 1)N + MR_f)$	$F_s(M + 1)(N - 1)$
Halfband filter	Power Efficient	Useful only for decimation by 2	$(N + 1)F_s/8$ for even $(N + 1)/2$ $(N + 3)F_s/8$ for odd $(N + 1)/2$	$(N + 1)F_s/4$

tasks were performed by ASIC accelerators while the DSP mainly performed the control operations [35]. The higher energy efficiency of dedicated hardware accelerators when compared to a software implementation, and the limited flexibility requirements justified such an approach. However this model becomes increasingly untenable when flexibility and multistandard support are introduced as major design constraints, in addition to a low power consumption. From the original ‘ASIC-centered DSP assisted’ model, baseband architectures for SDRs have been evolving in two different directions. In the first category are the class of architectures which use multiple Single Instruction Multiple Data (SIMD) units. Traditional DSP’s had a single multiply and accumulate unit in their datapath. Since they have to perform most of the processing sequentially, they would have to be clocked at very high frequencies to satisfy the high computational load of 3G and 4G standards, which in turn would result in a high dynamic power consumption. Most of the baseband algorithms are characterized by a large amount of data level parallelism. SIMD units exploit this inherent parallelism and the ability to vectorize the baseband kernels to amortize the cost of the instruction fetch and decode cycle over multiple data elements in parallel. An overview of the design choices made by the designers of the SIMD class of SDR baseband processors for implementing the digital front-end is given below.

General purpose processors have the highest amount of flexibility since their instruction set allows them to be reprogrammed to perform any arbitrary function. This generality comes at the expense of a low energy efficiency. Specializing the instruction set and datapath of the processors for a specific task in the form of an Application Specific Instruction Processor (ASIP) increases the energy efficiency by an order of magnitude [57, 58, 59]. CoreSonic’s LeoCore processor [12] exploits the above observation. It uses

2.5 Implementation of the DFE in existing SDR baseband architectures

Figure 2.19: Division of baseband load between 4 cores in LeoCore Processor

the fact that the algorithmic kernels in the baseband of most modern wireless standards, fall into one of four computing clusters. These kernels can in turn be allocated to four processor cores, each specialized for a different category of tasks as shown in Fig. 2.19. The four cores include a DFE engine, a complex SIMD processor, a set of functional accelerators and a control processor. Since the DFE signal processing tasks of filtering, pulse shaping and sample rate correction are algorithmically similar across different standards, the flexibility offered by an instruction set architecture is unnecessary, instead it is implemented as a parameterizable ASIC HW accelerator. The designers of both NXP's Embedded Vector Processor [15] and Infineon's MUSIC processor [11] have observed that the baseband kernels can be grouped into three computing domains: DFE, modem and codec. Among the three domains, the modem algorithms show the greatest amount of diversity between different standards, while the DFE and codec algorithms show a more limited variation between standard. In the EVP, the DFE is implemented using a configurable filter, while it is implemented using an ASIP. In the EVP, the modem algorithms are implemented on a Very Long Instruction Word (VLIW) capable SIMD vector processor. The VLIW capability allows parallelism across different vector functional units as well as scalar functional units. The MUSIC processor also implements the modem algorithms on a cluster of SIMD cores each consisting of four programmable elements (PE). The number of cores can vary depending on the processing requirements. The VLIW capability built into the cores is used for communication across the PEs. University of Michigan's SODA processor [13] splits the baseband processing tasks between a set of four data processing SIMD cores and a control processor. The data processing cores support both SIMD and scalar operations, and implement the computationally intensive DSP kernels while the control processor performs the data transfers to memory, and manages the data

2.5 Implementation of the DFE in existing SDR baseband architectures

Figure 2.20: Embedded Vector Processor

Figure 2.21: MUSIC Processor

processors. The EVP and MUSIC baseband architecture has been shown in Fig 2.20 and Fig. 2.21. In a departure from MUSIC and EVP, the SODA architecture favors scratchpad memories for inter-function communication rather than multithreading of the baseband kernels. It also differs from them in the fact that the SODA architecture does not use any dedicated HW accelerators and computationally intensive kernels, including the DFE tasks are completely implemented on the SIMD cores. The SODA baseband processor is shown in Fig. 2.22 Arm's ADBERG processor [13] is a commercial variant of the SODA processor. The main difference from SODA is the addition of HW

2.5 Implementation of the DFE in existing SDR baseband architectures

Figure 2.22: SODA processor

ASIC accelerator for Turbo decoding, block floating point operations and some arithmetic functions, which took a lot of processing cycles in the SODA architecture. The HW acceleration allows the frequency of the SIMD cores to be reduced. ADBERG also incorporates limited VLIW capability to handle some of the function level parallelism inherent in the baseband. Like SODA, the DFE computational load in the ADBERG processor is allocated to the SIMD cores. Performing the DFE computations on a HW ASIC accelerator or performing them in software represents a tradeoff between flexibility/upgradeability and a lower power consumption. Offloading the computations from the SIMD cores to an ASIC accelerator significantly reduces the power consumption, but it also potentially limits the ability to future-proof the baseband processor, due to the limited flexibility of the ASIC accelerator.

The second category of SDR baseband architecture evolution represents a more radical departure from the above DSP-centric baseband model, and is based on the use of reconfigurable arrays. The most commonly used reconfigurable array is the Field Programmable Gate Array (FPGA). FPGAs offer a fine-grained bit-level reconfigurability. Finer the granularity of the programmable elements in the array, higher is the overhead for interconnect. FPGAs typically have a very high power consumption compared to an ASIC due to a large amount of interconnect capacitance and a large amount of switching activity in the interconnect network. This makes them a poor candidate for

2.5 Implementation of the DFE in existing SDR baseband architectures

Figure 2.23: Montium tile

a energy constrained mobile handsets, despite its flexibility. Since most of the computationally intensive kernels in the baseband operate on word-level operators, a coarse grained reconfigurable array (CGRA) is a more suitable choice. Notable among this class of processors is Recore System's Montium processor [16]. The basic building unit comprises of a Montium tile which consists of five identical multi-level arithmetic and logic units (ALUs), with ten parallel local memories providing a large memory bandwidth, as shown in Fig. 2.23. The ALUs are optimized for computationally intensive 16-bit kernels. Each ALU contains two levels: Level 1 contains generic arithmetic and logic functions, while Level 2 contains a dedicated MAC unit for filtering and FFT operations. Either level can be easily bypassed. Two simple controllers are responsible for sequencing the operations within the tile and for managing the communications outside the tile. The MONTIUM processor is more closer to an FPGA than a DSP processor in that, once it is configured by a bitstream, there are no instruction fetch and decode cycles during the operation. For implementing the entire baseband load, an array of MONTIUM tiles can be designed, where the number of tiles is a function of the targeted computational load. The DFE computational load can also be allocation to the MONTIUM tiles. The computational efficiency of CGRAs for dataflow operations, and the relative ease of programming offered by DSPs have also inspired hybrid approaches like IMEC's ADRES processor [10]. This processor combines three VLIW functional units, a predicate register file and a CGRA comprising of sixteen interconnected PEs,

2.5 Implementation of the DFE in existing SDR baseband architectures

Figure 2.24: ADRES baseband core

as shown in Fig. 2.24 It can dynamically switch between the VLIW mode and CGRA mode at runtime, while switching between a dataflow task and a control task. The absence of instruction fetch overheads decode make a CGRA style implementations more power efficient than a SIMD style implementation of the DFE. However the generic nature of CGRA PE in the MONTIUM or the ADRES processor still makes them less power efficient than a dedicated ASIC HW accelerator for performing the DFE tasks. Toshiba's MT-ADRES [17] addresses this problem by extending the ADRES core with dedicated HW accelerators for the DFE and the Forward Error Correction (FEC) decoder. The ADRES core performs only the modem algorithm. The modem algorithms being very diverse across different standards, better exploit the flexibility offered by the ADRES core. This strategy is similar to the one adopted by the EVP and MUSIC processors. Intel's Scalable Communications Core (SCC) [18] can be considered as a variation of the CGRA, the fundamental difference being that the PEs in the above array are heterogeneous coarse grained accelerators, each of them optimized for a specific class of baseband kernel tasks. The SCC handles only the modem and codec algorithms, while the DFE tasks are handled by a dedicated accelerator outside the core. Homogeneous arrays like MONTIUM and ADRES have a simpler programming model, but are less efficient for handling diverse categories of baseband kernels. The DFE implementation strategy in the different baseband architectures are summarized in Table 2.3.

2.6 Limitations of Existing Decimation Filter Designs for SDR Channelization

Table 2.3: DFE Implementation in SDR Baseband Architectures

SDR Baseband Processor	Design Style	DFE Impementation
LeoCore [12]	ASIP	HW Accelerator
MUSIC [11]	SIMD	HW Accelerator
EVP [15]	SIMD	HW Accelerator
SODA [13]	SIMD	SIMD core
ADBERG [13]	SIMD	SIMD core
MONTIUM [16]	CGRA	CGRA
ADRES [10]	VLIW-CGRA	CGRA
MT-ADRES [17]	VLIW-CGRA	HW Accelerator
SCC [18]	Heterogeneous CGRA	HW Accelerator

2.6 Limitations of Existing Decimation Filter Designs for SDR Channelization

Details of the DFE implementation in most of the SDR processor designs discussed in the last section are not readily available in the public domain. A survey of the DFE literature reveals a large number of filter designs capable of performing the channelization function, either partly or wholly. These designs are mostly stand-alone HW accelerators not coupled to any specific SDR baseband processor. Typically they assume that the channel of interest has been shifted to DC, either through analog quadrature down-conversion stage or a NCO. A survey of the reported designs reveals that they broadly fall under one of the following two categories:

- Multistage decimation filters for $\Sigma\Delta$ ADCs.
- General architectures for sample rate conversion.

A good candidate architecture for SDR channelization requires the following criteria to be satisfied:

- Functional completeness - the designs should perform all the required DFE tasks of sample rate conversion, interferer attenuation and pulse shaping secondly the architecture should reusable or reprogrammable for supporting different standards.

2.6 Limitations of Existing Decimation Filter Designs for SDR Channelization

- Reusability and Upgradeability- the designs should ideally have a high degree of hardware reuse between standards and should be capable of being upgraded to support a new standard.

In addition to sample rate conversion and noise/interference attenuation, the decimation filter chain in the DFE also has the dual role of channel selection. This has not been factored into majority of $\Sigma\Delta$ ADC decimation filter designs. Channel selection and decimating to the symbol rate requires a pulse shaping filter to alleviate the effect of ISI and minimize symbol estimation errors. The oversampling factor is one of the design parameters of pulse shaping filters. Hence while partitioning the original SRC factor into multiple stages, the oversampling factor of the pulse shaped filtering stage also has to be taken into account. A digital RRC filter needs an input OSR of at least two [54]. Only a few works like [76, 77] have taken the pulse shaping filter into account while partitioning the total decimation load.

The modem algorithm after the DFE operates at the symbol rate. However most of the reported decimation filter designs have focused only on the integral decimation between $\Sigma\Delta$ ADC sampling rate and the Nyquist rate. These filters usually assume that any subsequent fractional sample rate conversion between the Nyquist rate and the symbol rate can be performed efficiently by a resampler operating in a low OSR environment after the integral decimation chain [78, 80, 81, 83]. While the interferers and blockers attenuation requirements progressively decrease down the decimation chain, a fractional SRC stage operating in a high OSR environment needs to attenuate potential aliasing components only within a very narrow band. The stringency of the transition bandwidth specifications progressively increases in a multistage decimation filtering chain. Hence a fractional SRC operating in a relatively high OSR environment can usually be implemented in a much simpler fashion than ones operating in a low OSR environment [64]. In [64], the authors use a modification of the CIC filter called the time-varying CIC filter to perform a fractional decimation in the first stage, followed by subsequent integral decimation stages. The lack of multipliers operating at the $\Sigma\Delta$ ADC sampling rate was one of the attractions of the CIC filter. The main drawback of the time-varying CIC structure is the introduction of multipliers into the CIC structure which result in a high power consumption.

A few multistandard $\Sigma\Delta$ ADC decimation filter designs decimate the ADC sampling rate to the symbol rate. In [62], the authors suggest that the $\Sigma\Delta$ ADC sampling rate

2.6 Limitations of Existing Decimation Filter Designs for SDR Channelization

to be an integral multiple of the standard specific symbol rate. This allows the required decimation to the symbol rate to be performed entirely by integral decimation stages. Constraining the ADC sampling rate can eliminate the requirement of a fractional SRC stage, but it limits the flexibility of the radio to a set of standards whose symbol rates are known a priori. Any new standard, whose symbol rate is not an integral factor of the fixed clock cannot be supported. Due to the difficulty of providing multiple high quality master clocks [64], it can be envisioned that the total number of available master clocks in a multistandard radio would be limited. In [87], a fractional SRC stage is incorporated into the decimation stage. This allows any arbitrary integral or rational decimation factor to be factorized into the form $M \times 2^K / \mu$. M represents a variable integral non-prime factor that is performed using a CIC stage. μ is fractional interpolation factor performed by a subsequent fractional SRC stage. The factor 2^K is performed by K halfband stages.

The decimation filter chain has the task of decimating the highly oversampled input from the $\Sigma\Delta$ ADC to the symbol/chip rate. The task is not merely a simple sample rate conversion operation, due to the presence of blockers, interferers, and quantization noise in the input spectrum. Hence each stage needs to ensure that the aliasing components for that particular stage are attenuated as per the C/N requirements of the current mode of operation. The design of a multistandard decimation chain is complicated by the fact that, the band edge specification of the individual stages, the position of the aliasing components, their strengths as well as the C/N of the standards can show considerable variation [62], thereby requiring each of the above stages to be programmable. Hence the multiplication operations for these stages have to be performed using general multipliers. The need for programmability precludes the use of the vast body of existing literature on low-power, low-area optimizations for constant-coefficient FIR filters, which could have been potentially used in the channelization accelerators for single mode radio.

In [82], the authors propose the use of a Residue Number System (RNS) based decimation filters for capable of supporting the WCDMA and IEEE 802.11a standards. The underlying idea is to reduce the power consumption of the reprogrammable filter stages when generic MAC units are used. RNS based MAC units have a shorter carry propagation and hence have a lower amount of switching activity than traditional MAC architectures. The filter stages are implemented in full-parallel style and the number

2.6 Limitations of Existing Decimation Filter Designs for SDR Channelization

of programmable MAC units allocated to each stage depends on the standard with the more stringent requirement.

Not many existing works have focused on the issue of identifying redundancies that may exist across multiple standards in a multistandard multistage decimation filter, which may allow one or more stages to be hardwired and reused across standards with minimal changes. If a filter stage can be designed such that it can be hardwired, the coefficient-multiply operations can be performed by a network of hardwired shifts and additions, resulting in significantly lower power and area than a general multiplier based full-parallel transpose form FIR structure [23]. Hence the programmability overheads for multistandard support in channelization accelerators, translate to a much higher power consumption and area than their single mode counterpart. Traditional recursive CIC filters have a high degree of reusability, since varying decimation rates can be implemented by merely changing the downsampler clock, while the coefficient-less IIR and FIR stages can be directly reused without any changes. However, the passband droop and non-uniform attenuation in the passband makes it suitable only for decimation at a high OSR, and the subsequent decimations needs to be handled by a cascade of halfband filters, CIC-compensation filters and fractional SRC stages. These stages cannot be readily reused because of standard dependent bandedges and stopband attenuation parameters. The few works that have touched upon the issue of filter stage reusability have limited their scope to a very small set of standards, hence the results are not easily extendible to a multistandard radio, that is capable of supporting arbitrary number of standards having arbitrary specifications.

Most of the reported decimation filter designs which handle only integral decimation loads, partition the above load between a CIC stage, a halfband stage and FIR stage [78, 80, 81, 83, 84], where the halfband and last stage FIR filter decimate by a factor of 2. In [85] the authors propose a cascade of a CIC stage, two halfband stages and a FIR stage for the DFE. The CIC stage and the last stage filter support programmable integral decimation factors. Both [85] and [84] suggest that the halfband filter can be hardwired and reused between the different standards. The architecture for arbitrary sample rate conversion in [88] also reuses multiple halfband filter across different standards. A hardwired filter can be implemented in a multiplierless fashion using constant multipliers. However these designs do not discuss the design parameters of the hardwired halfband filter. The passband edge (in radians) that the halfband filter needs to

2.6 Limitations of Existing Decimation Filter Designs for SDR Channelization

protect from aliasing is a function of both the information bandwidth of the particular standard and the OSR environment in which the halfband filter operates. A halfband filter designed for a specific passband edge cannot prevent aliasing when reused for a standard that has a wider passband edge requirement. This may act as a bottleneck for future upgradeability.

Some decimation filter designs have also targeted a multiplierless implementation of a decimation stage by constraining the filter to have only unity coefficients. In [86] the authors propose a integral decimation filter structure that consists of a cascade of a specific class of cyclotomic polynomial (CP) filters having only unity coefficients. The design procedure exploits the fact that the potential aliasing components of single decimation stage lie within a small folding band which lies around integral multiples of the output sample rate of the decimation stage. Suitable CPs which have zeroes at the potential aliasing band, and no zeros in the passband are identified. The authors then use an optimization procedure to find the optimum linear combination of the chosen CPs which satisfies the design constraints. This method can be used for efficient hardwired implementation of a decimation stage. The main drawback of the method is that designing for a different decimation factor, will results in a different set of candidate CPs, and a totally different architecture after the optimization procedure. Hence the strategy is not suitable for a decimation stage that has to support variable integral decimation factors.

A specific class of decimation stage in [89, 90], to tackle the above problem of designing a decimation stage that can support variable decimation factors. Implementing such a stage using a generic FIR filter, would require the entire filter to be reprogrammed which switching between standards. The structure in [89] is based on the Farrow's structure and has fixed subfilters. Reprogramming the structure to support a different integral decimation factor requires the tuning of a single parameter. The structure can support arbitrary integral factors (including prime factors). Recently the above structure was also extended for integral interpolation or decimation of a bandpass signal [90]. The decimation structure in [89] could potentially be used in situations where an prime decimation factor cannot be factorized into multiple integral decimation stages. Note that when a fractional SRC is present in the decimation chain, one has more flexibility to factorize any arbitrary integral or rational factor and more opportunity to

use specialized structures like non-recursive CIC filter filters or halfband filters in the multistage decimation chain.

2.7 Conclusion

In this chapter, we gave an overview of the DFE, its functional tasks, and the challenges that exist in designing and implementing a DFE for a multistandard SDR handset. Evolution of the wireless standards towards 4G has dramatically increased the computational loads in the digital baseband. The higher throughput requirements have resulted in increased channel bandwidths. Reducing the flexibility requirements of the analog front-end requires the bulk of the channelization load to be pushed to the digital domain. The high computational load of the channelization function in the DFE precludes a software implementation in an energy constrained mobile handset. Instead, the function needs to be implemented using an accelerator. HW accelerators are typically optimized for a single specification. In a multistandard SDR, the channelization accelerator has to accommodate varying channel bandwidths, passband and stopband edges, interferer attenuation requirements, SRC factors and pulse shaping factors. These variable channelization parameters impose a high degree of flexibility on the accelerator. A viable channelization accelerator design strategy should ensure that area and power overheads of flexibility do not violate the energy and resource constraints of a mobile handset.

3

Flexibility Penalty and Hardware Reuse in Flexible Filter Accelerators

The survey of the DFE implementations in existing SDR baseband architecture, discussed in Section 2.5 reveals two distinct design strategies. Homogeneous multiprocessors like SODA, ADRES and MONTIUM implement all the computationally intensive baseband kernels on an array of identical PEs. Such approaches have the advantage of having a simpler programming model. The regularity allows an efficient silicon implementation and the use of a regular interconnect network. However the PEs are designed to be sufficiently general to accommodate a wide variety of baseband kernels. For instance, the PE has to perform the computationally intensive DFE filtering tasks, which has a lot of inherent data level parallelism and can be easily vectorized. On the other hand it also has to accommodate the computationally intensive codec algorithms which are difficult to vectorize. This lack of specialization necessitates increased number of PEs, higher operating frequencies, and a high power consumption.

Heterogeneous multiprocessors like LeoCore, SCC, MT-ADRES, EVP and MUSIC offer more scope for specializing the PEs for performing a specific class of baseband kernels, which translate to a lower area and power consumption, which are the critical constraints in a mobile handset. These benefits come at the cost of a few drawbacks. They typically have a more complex interconnect network. A more critical problem is the risk of obsolescence. Specialized units can act as a bottleneck for future in-field

upgradeability, which is one of the important goals envisioned for a SDR. Redesigning, verification and testing of an ASIC HW accelerator can take upto an year, which increases the time to market, when the baseband processor needs to be upgraded to support a new standard. Hence flexibility is a critical design requirement in the design of specialized HW accelerators for computationally intensive kernels like the DFE. In the traditional ‘ASIC centered DSP assisted’ baseband model [35], the HW accelerators are inflexible since they are optimized for a single specification. When the specifications are fixed, the ASIC design flow (shown in Fig. 3.1) can exploit a large class of high level synthesis behavioral optimizations like common sub-expression elimination, constant propagation, operator strength reduction, loop unrolling etc., which results in significant reduction area and power [91].

Figure 3.1: Single mode ASIC channelization accelerator design flow.

Flexibility precludes the use of behavioral optimizations which exploit the presence of constant input parameters in the accelerator architecture, and necessitates the use of generic hardware. Thus flexible accelerators incur a power and area penalty. An additional dimension that is important in flexible accelerators is the hardware reconfiguration latency, which is the time taken to parameterize the accelerator for a new mode. In this work, the area, power, and reconfiguration latency overheads incurred by an accelerator, in order to meet the flexibility requirement for supporting multiple standards, are referred to as the ‘flexibility penalty’. The relative positions of single

mode and flexible accelerators in the accelerator design space are illustrated in Fig. 3.2.

Figure 3.2: The flexible accelerator design space and the flexibility penalty.

Assuming that the channel of interest has been downverted to DC, the principle DFE channelization tasks of channel selection, interferer attenuation, pulse shaping and sample rate conversion can be reduced to a set of filtering operations. The filter accelerator needs to be flexible for accomodating varying channel bandwidths, band-edges, interferer locations and power levels, and SRC factors. The area, power and reconfiguration latency penalties are closely related to the granularity of hardware reuse across multiple standards. Reusing hardware at coarser levels of granularity offers more opportunities for power optimizations, reduces the area overhead when a large number of standards need to be supported, and reduces the amount of reconfiguration data needed to parameterize the accelerator. In this chapter, we analyze the area, power and reconfiguration latency components of the flexibility penalty, as well as the relationship between the flexibility penalty and the granularity of hardware reuse in the existing flexible filter accelerator paradigms.

3.1 The Flexibility Penalty

A wide variety of design styles exists for implementing a flexible filter HW accelerator in the DFE of a SDR. These styles differ in the granularity of hardware reuse, the degree of hardware multiplexing, and the use of low level area/power optimizations. Hence they offer varying tradeoffs between the time taken to reconfigure the accelerator for new mode, the area and the power consumption. This section gives a detailed look at factors which contribute to the the power consumption, area overheads and reconfiguration time penalties, as well as their practical repercussions of in a mobile SDR handset.

3.1.1 Power Consumption

Flexible hardware platforms are always less power efficient than custom hardware optimized for a single mode [9]. In energy constrained battery powered handsets, the power consumption is directly linked to the usable battery life of the device. Traditionally, the active power consumption of the device was dominated by dynamic power. Mobile devices typically spend a significant amount of time in standby mode, during which leakage power is more dominant. With scaling of CMOS technologies into the nanoscale era, the contribution of the leakage power is becoming comparable to that of the dynamic power even during the active modes, which necessitates a re-look at the traditional power optimization strategies. The evolution of CMOS technologies has more or less followed Moore's law which predicts that the density of transistors would double, every 24 months [92]. This requires the device dimensions to be scaled by a factor of 0.7 in each successive generation. In the constant field scaling regime, the lateral and vertical device dimensions, the supply voltage and the threshold voltage are all scaled by a factor of 0.7, whereas the doping density is scaled by a factor of $1/0.7$. This allows the magnitude of the electric field in the gate oxide and the MOSFET inversion channel to remain more or less unchanged. The main attraction of technology scaling lies in the fact that both the delay and the capacitance are scaled by a factor of 0.7. Hence the maximum operating frequency of the transistors is increased by about 43%, while the scaled capacitance and supply voltage reduce the switching energy by a factor of 0.73 for each generation [93].

For more than three decades, aggressive constant field scaling has led to increased performance, reduced dynamic power consumption and increased transistor density.

3.1 The Flexibility Penalty

However, continued scaling has become unsustainable in the nanoscale era due to the increased role of leakage power. In higher device geometries, leakage was considered negligible in comparison to dynamic power, and was an important consideration only for portable battery-powered devices spending a significant amount of time in the standby mode. Active power consumption was always dominated by dynamic power. However, the leakage power contribution exceeds the dynamic power consumption in the sub-65nm geometries. This behavior can be attributed to two main factors. Firstly, scaling of the supply voltages means that the threshold voltage must also be scaled down to maintain the gate overdrive. Reducing the threshold voltage causes an exponential increase in the subthreshold leakage current [94]. Secondly, scaling of the silicon dioxide gate oxide causes the gate and the channel to be separated by the thickness of just a few atoms, and this effect tremendously increases the gate tunneling currents. The cumulative effect of these two currents has resulted in the leakage power overtaking the dynamic power as the dominant source of power consumption in nanoscale CMOS circuits. Fig. 3.3 shows the relative contribution of the major power consumption components over the years [95]. The nanoscale CMOS power consumption components are discussed in more detail below.

Figure 3.3: CMOS power consumption trends.

3.1.1.1 Subthreshold Leakage Power

The leakage mechanisms that exist in nanoscale CMOS circuits are shown in Fig. 3.4. The subthreshold leakage current refers to the drain-to-source current that flows when the gate-to-source voltage, V_{GS} , of the transistor is below the threshold voltage. Under this condition, the transistor is said to be in a weak inversion mode. The current in the weak inversion mode is dominated by the diffusion current, rather than the drift current which dominates the drain-to-source current in a strong inversion mode. This behavior can be attributed to the low concentration of minority carriers and the smaller longitudinal electric fields. In long channel transistors, electrons are attracted to the

Figure 3.4: Leakage currents in a MOSFET.

channel from a highly doped source, under the effect of a positive gate voltage. With increased scaling of the channel length, the drain and source depletion regions are very close to each other. This close proximity causes the electrons in the source to be pulled into the channel under the effect of the drain electric field. Hence, in short channel transistors, the drain bias, V_{DS} , influences the channel formation as well as the gate voltage. The resultant lowering of the potential barrier for electrons in the source to enter channel reduces the threshold voltage, and the effect is called drain-induced barrier lowering (DIBL) [94].

Another important factor that affects the threshold voltage is the source-body bias. For a MOSFET to function properly, both the source-body p-n junctions and the drain-body p-n junctions have to be reverse biased. Electrons are attracted to the

3.1 The Flexibility Penalty

channel from the source whenever the channel depletion width is comparable to that of the source-body depletion width. At this point, the electrostatic potential of the channel surface is equal to that of the source region. Increasing the magnitude of the reverse-biased source-to-body voltage, V_{SB} , results in the widening of the source-body depletion region. Consequently, an increased gate voltage must be applied to further increase the channel depletion width. The dependence of the threshold voltage, V_{th} , on the body bias, V_{SB} , is modeled as follows [94]:

$$V_{th} = V_{FB} + 2\psi_B + \frac{\sqrt{2\epsilon_{si}qN_A(2\psi_B + V_{SB})}}{C_{ox}} \quad (3.1)$$

where V_{FB} is the flatband voltage, N_A is the doping density, C_{ox} is the gate oxide capacitance per unit area, ϵ_{si} is the permittivity of silicon, q is the electronic charge and ψ_B is the difference between the Fermi potential and substrate intrinsic potential. The cumulative effect of DIBL and body bias can be incorporated into Eq. 5.4 as follows [94]:

$$V_{th} = V_{FB} + 2\psi_B + \frac{\sqrt{2\epsilon_{si}qN_A(2\psi_B + V_{SB})}}{C_{ox}} - \eta V_{DS} \quad (3.2)$$

where η is the DIBL coefficient. When the body bias and DIBL are taken into consideration, the substrate leakage power can be modeled as follows [94, 96].

$$P_{sub} = DV_{DD}e^{\left(\frac{V_{GS}-V_{th}}{mv_T}\right)} \left(1 - e^{\left(\frac{-V_{DS}}{v_T}\right)}\right) \quad (3.3)$$

where D is given as:

$$D = \mu_0 C_{ox} \frac{W}{L} (m-1) v_T^2 \quad (3.4)$$

V_{DD} is the supply voltage, m is the body effect coefficient, v_T is the thermal voltage, μ_0 is the zero bias mobility, W is the gate width and L is the gate length. The body effect coefficient, m , is expressed as follows:

$$m = 1 + \frac{3T_{ox}}{W_{dm}} \quad (3.5)$$

3.1 The Flexibility Penalty

where W_{dm} is the maximum depletion layer width, while T_{ox} is the thickness of the SiO_2 gate oxide. Eq 3.3 indicates an exponential relation between the subthreshold leakage power and the threshold voltage, and a linear relationship with the gate width.

3.1.1.2 Gate Leakage Power

In an n-channel transistor, the n+ doped polycrystalline silicon gate is separated from the n-type channel by an insulating layer of silicon dioxide (SiO_2), which is assumed to have an infinite impedance. However, aggressive scaling has resulted in the gate oxide approaching a thickness of approximately 1.2 nm or, equivalently, 5 silicon atoms. At this thickness, the electrons can directly tunnel through an extremely narrow potential barrier, resulting in gate leakage currents [94, 97]. A reverse situation exists in the off state ($V_{GS} = 0$, $V_{DS}=V_{DD}$), when the electrons in the n+ polycrystalline gate tunnel directly into the drain extension regions. Fig. 3.4 shows both the mechanisms. Similar hole tunneling currents also exist in p-channel transistors. The power consumption due to the direct tunneling gate leakage current can be modeled as follows [98, 99]:

$$P_{gate} = AV_{DD}WL \left(\frac{V_{ox}}{T_{ox}} \right)^2 e^{\left(-B \left[\frac{1-(1-V_{ox}/\phi_B)^{1.5}}{V_{ox}/T_{ox}} \right] \right)} \quad (3.6)$$

where ϕ_B is the barrier height, V_{ox} is the voltage drop across the gate oxide and the constants A and B are functions of the effective carrier mass and the barrier height ϕ_B . A closed-form expression for V_{ox} in terms of the gate voltage, the Fermi level and the flatband voltage can be found in [98]. The expression (6) shows that the gate leakage power has an exponential dependence on the supply voltage and the thickness of the gate oxide, and a linear dependence on the gate width.

3.1.1.3 Dynamic Power

Dynamic power consumption occurs when switching activity induces charging and discharging of capacitive nodes. The switching activity comprises of both the useful activity that is necessary for evaluating a particular output and the spurious switching activity that a node may experience before settling into a steady state value. The load capacitance typically includes both parasitic transistor capacitance and interconnect

3.1 The Flexibility Penalty

capacitance. In nanoscale CMOS technologies, the latter component has started dominating due to the non-scaling of the interconnect capacitance. The dynamic power consumption can be expressed as follows [100]:

$$P_{dyn} = \alpha_f C_T V_{DD}^2 f_{clk} \quad (3.7)$$

where C_T stands for the total capacitive load, α_f is the fraction of capacitive nodes that are switching and f_{clk} is the clock frequency. Eq. 3.7 reveals a quadratic dependence on the supply voltage, and a linear dependence on the effective capacitance and frequency.

3.1.2 Area

Area efficiency is a requirement which is shared by all the computationally intensive kernels in the multistandard radio baseband. Instruction set architectures, at the expense of a lower power efficiency, offer a high degree of reusability since the same underlying hardware can be used for implementing arbitrary functions defined in software. Shifting a function from software to hardware in the traditional ‘ASIC centered DSP assisted’ baseband model carries the risk of making the design unscalable in a multistandard radio due to the limited flexibility/reusability of the hardware accelerators. When a large number of standards need to be supported, using dedicated hardware accelerators to implement the computationally intensive kernels in all the standards can translate to a very large area penalty, even if each of the accelerators individually have a high performance-to-area ratio. For very high volume products like mobile handsets, keeping the silicon die area to a minimum has a very big impact on the costs. Smaller chip area allows more dies to be packed in a single wafer resulting in higher yields [101].

For battery powered mobile devices which spend a significant amount of time in standby mode, the leakage power consumption is a very important design constraint. Both gate leakage and subthreshold leakage power consumption are strongly correlated to the total gate width and area [95]. Hence reducing the area and the number of leaking transistors is important to reduce the overall standby power consumption.

Aggressive scaling over the years has also increased the gate density, allowing more number of gates to be packed within the same die area. Hence a naive approach for designing multistandard accelerators would be to exploit the increased gate density by implementing multiple standard specific accelerators in parallel. The dynamic and

3.1 The Flexibility Penalty

leakage power of the unused accelerators could potentially be reduced by clock gating and power gating respectively. While clock gating significantly reduces the switching activity, power gating as a leakage reduction strategy does not scale well into nanoscale technologies due to gate leakage currents in the added power gating circuitry [102]. With the scaling of the gate oxide thickness, the gate leakage power has been increasing at a much faster rate than the subthreshold leakage. The above brute force approach is not power efficient for nanoscale technology nodes. Thus, area efficiency and reusability of all the hardware accelerator cores are important from the perspective of scalability, manufacturing cost reduction and standby leakage reduction.

3.1.3 Hardware Reconfiguration Latency

In a heterogeneous multiprocessor implementation of the SDR baseband, the computationally intensive kernels are performed in HW accelerators, while a reduced instruction set computer (RISC) controller typically manages the data flow and is responsible for parameterizing the accelerators for a new mode. The configuration contexts for the new mode can be assumed to be either stored in a global memory or supplied through an over-the-air update. The hardware reconfiguration latency primarily arises due to the finite memory bandwidth of the processor managing the terminal reconfiguration. The processor requires a large number of clock cycles when a large amount of reconfiguration data has to be transferred from the global memory to the local memories and configuration registers of the parameterizable accelerator cores while switching to a new mode. The hardware reconfiguration latency thus represents the cumulative time taken to reconfigure all the physical layer functional units. Parameterizable designs in general are typically not optimized for on-the-fly reconfiguration, as reconfiguration is usually assumed to be an offline process. However emerging applications like CRs require the physical layer to be extremely flexible to rapidly respond to a changing radio environment, and have to switch between multiple modes at a much faster rate than traditional multistandard radios. Hence it is important to ensure that introducing the flexibility needed for multimode support in the hardware accelerators does not incur significant penalties in terms of the amount of reconfiguration data required to parameterize the accelerator for a new mode.

The impact of a high reconfiguration latency in an opportunistic spectrum access scenario is analyzed below. The traditional spectrum licensing model provides the licensed

3.1 The Flexibility Penalty

primary users exclusive access to a portion of the spectrum. The usage of the licensed spectrum may be incomplete in time and space, leading to spectrum holes which can be exploited in an opportunistic spectrum access model [3]. However such a model should ensure that the interference experienced by the primary users is minimized. On detecting the presence of a primary user, the spectrum-aware cognitive terminal should immediately vacate the current channel and perform its transmission activities over a new vacant channel seamlessly. This requires the cognitive terminal to have both flexibility and spectrum awareness. Flexibility is required to optimize the link parameters for any vacant channel while spectrum awareness is required to detect the presence of primary users and spectrum holes. Minimizing the interference to the primary users, ideally requires the spectrum to be sensed continuously. However this is likely to increase the number of false alarms due to the transmitted signal being interpreted as a primary user signal. This is the underlying rationale behind the periodic sensing structure, in which sensing and transmission are performed in mutually exclusive time slots [103]. Optimum scheduling of the sensing and transmission time slots represents a tradeoff between interference minimization and spectrum efficiency. Increasing the sensing time allows the terminal to be more responsive to the presence of a primary user in order to reduce any potential interference. However doing so reduces the transmission time and hence the system throughput. Increasing the transmission intervals increases the spectrum efficiency but also results in increased probability of interference to primary users, due to the absence of any sensing data in the above intervals.

During the sensing intervals, the spectrum is sensed for the presence of primary users, and by the end of the interval a decision is taken, whether to continue transmitting on the same channel or to switch to a new spectrum hole. The sensing decision can be made either locally or in cooperation with the other cognitive peer nodes [104]. A locally taken decision suffers from the hidden node problem and increased probability of failure to detect primary users due to channel fading. A cooperative sensing model avoids the above drawbacks by fusing several locally made decisions in a fusion node [105, 106]. By the end of the sensing interval, the decision is communicated to both the transmitting and receiving cognitive nodes. Assuming that the sensing intervals are of duration T_S and the transmit intervals are of duration T_X , the sensing efficiency, E ,

3.1 The Flexibility Penalty

can be defined as follows [103].

$$E = \frac{T_X}{T_X + T_S} \quad (3.8)$$

The sensing efficiency gives a measure of the proportion of the total time available for transmission. Since the throughput is strongly dependent on the transmission time, the sensing efficiency also gives an indication of the achievable system throughput. Eq. 3.8 however does not take into account the finite amount of delay incurred in reconfiguring the hardware accelerators, in case a decision is made to switch to a new channel. The effect of the reconfiguration latency can be modeled as follows. The occupancy of the primary user in a given channel can be assumed to be an on-off process with independent and identically distributed (i.i.d.) on-off periods. The on and off periods correspond to the occupied intervals and vacant intervals of the channel respectively. These periods are assumed to have an exponential distribution with mean values of T_{on} and T_{off} [107] respectively. To simplify the analysis, all potential channels are assumed to have similar occupancy characteristics. Under these assumptions, the probability of the current channel being occupied in a given sensing interval is given as follows [107].

$$p = \frac{T_{on}}{T_{on} + T_{off}} \quad (3.9)$$

It is assumed that it is always possible for the cognitive terminal to find an alternate vacant channel in case the current channel is occupied. Let the cumulative time taken to reconfigure all the accelerators in a cognitive terminal be ΔT . When two communicating peer nodes switch to a new channel, the practically available transmission time is only $T_X - \Delta T$, as indicated in Fig. 3.5.

The average transmission time can be evaluated as follows:

$$T_{avg} = (1 - p)T_X + p(T_X - \Delta T) = T_X - p\Delta T \quad (3.10)$$

The sensing efficiency can be reformulated to include the effect of the reconfiguration latency as follows:

$$E = \frac{T_X - p\Delta T}{T_X - p\Delta T + T_S} = \frac{1}{1 + \frac{T_S}{T_X - p\Delta T}} \quad (3.11)$$

Eq. 3.11 reveals that for a given transmission and sensing interval, an increase in the reconfiguration latency ΔT or the channel occupancy probability p can result in a lower sensing efficiency and hence a lower throughput.

3.2 Design Paradigms for Flexible Filter Accelerators

Figure 3.5: Periodic sensing structure.

3.2 Design Paradigms for Flexible Filter Accelerators

This section studies various flexible filter architectures with regard to their ability to provide multimode support and their flexibility penalty.

3.2.1 Velcro Approach

Figure 3.6: Velcro style accelerator.

When a coarse band consisting of several channels is digitized by the ADC, the input sample rate of the digital front-end is highly oversampled with respect to the symbol

3.2 Design Paradigms for Flexible Filter Accelerators

rate. Filtering the oversampled input is extremely computationally intensive task due to the large number of multiply operations per second. A decimation filter is usually implemented in multiple stages. This allows relatively lower order filters to be used at the high sampling rates in the earlier stages while allowing higher order filters to be used at lower sampling rates in the following stages, effectively reducing the overall multiply operations per second [61]. A single mode multistage decimation filter can benefit from further reductions in the arithmetic complexity during the high level synthesis, since the filter coefficients are constant [91]. The use of constant-coefficient behavioral optimizations like coefficient optimization, constant propagation and operator strength reduction reduce the multiply operations to a set of shift and add operations. When the filter stages are implemented in transpose direct form, common subexpression elimination (CSE) and graph dependency optimizations can be used to eliminate redundancies across multiple coefficients and reduce the adder count [10-14]. After the application of the above optimizations the resulting control data flow graph (CDFG) of the filter is highly irregular and can show considerable variations across the multiple modes, which makes hardware resource sharing across multiple modes, a non trivial task.

The simplest strategy to benefit from the reduction in arithmetic complexity offered by constant-coefficient behavioral optimizations while still providing multimode support would be a Velcro like approach [6], as shown in Fig. 3.6, where multiple single mode filters are instantiated in parallel, each of them optimized for a single mode. The obvious disadvantage of this approach is its non-scalability. When a large number of modes have to be supported, instantiating multiple filters incurs a large on-chip area in the baseband MPSoC. In nanoscale CMOS, leakage power is the most important source of power dissipation [95]. Leakage power is strongly dependent on the number of leaking transistors and hence the area. Hence the Velcro approach incurs a significant area and leakage power penalty. The flexibility offered by the above approach is also limited in that it assumes all possible modes to be known a priori. Hence it cannot support potential in-field upgrades for supporting a new mode. It however has the advantage of having a low reconfiguration overhead. Switching between multiple modes requires a single multiplexer and a demultiplexer to select the filter corresponding to the current mode, as shown in Fig. 3.6. Assuming N modes have to supported, a control word of size $\lceil \log_2(N) \rceil$ is sufficient to parameterize the multimode accelerator.

3.2 Design Paradigms for Flexible Filter Accelerators

3.2.2 Reconfigurable Datapath with Coarse Grained Functional Units

Figure 3.7: Synthesis of Multimode ASICs using Graph Merging.

The high area and leakage power overhead of the Velcro approach stems from the fact that it maps time-wise mutually exclusive modes onto different spatial functional blocks, even though only one of them is used at any point of time. An alternate approach to exploit the reduction in arithmetic complexity offered by constant coefficient optimizations, while avoiding the spatial redundancy of the Velcro approach would be to use graph merging approaches for synthesizing a reconfigurable datapath [86-90]. In the above class of methods, the control data flow graphs (CDFGs) of the multiple modes after the high level synthesis are merged into a single graph as shown in Fig. 3.7. The binding, scheduling, allocation and architectural synthesis stages are performed on the merged graph and are common to all the stages, resulting in a single reconfigurable datapath consisting of coarse grained hardware functional units that can support multiple modes by supplying suitable control signals to the steering logic. These approaches exploit the fact that hardware resources used in time-wise exclusive modes can be shared, resulting in lower datapath area. Fig. 3.8 shows CDFG of a constant multiplier. The vertices in the CDFG correspond to data operations while

3.2 Design Paradigms for Flexible Filter Accelerators

the arcs correspond to a dataflow between the data operations. The arcs may also be

Figure 3.8: Portion of a Filter CDFG.

associated with a hardwired left or right shift. Shifted arcs have been annotated by their corresponding weight. It is assumed that the high throughput rates in the digital front-end would necessitate a full parallel implementation with each vertex mapped to a dedicated hardware functional unit. Hence there is no intra-mode sharing of hardware resources. Each vertex in the merged CDFG represents a mapping from multiple vertices, each of them from different CDFGs. Each input port in a vertex of the merged CDFG may be associated with multiple input arcs, hence a multiplexer would be required to select the appropriate arc in a given operational mode. The datapath area of the merged design is close to that of the mode with the highest datapath area. However the additional steering logic required in this approach incurs an area and power penalty. The merging algorithms exploit subgraph isomorphism to reduce the amount of steering logic required [111]. Like the Velcro approach, the flexibility offered by the graph merging approaches is also limited to a known set of modes, and are incapable of supporting any field upgrades.

Assuming that N modes are merged, each hardware functional unit is shared by at most N modes. This implies that the maximum number of unique input arcs in an input port of a shared functional unit is also N . CDFG merging is illustrated for a simple example of three integer constant multipliers in Fig. 3.9 [113]. Hence a binary word of size $\lceil \log_2(N) \rceil$ is sufficient to act as the control signal for any multiplexer instantiated in the merged design.

3.2 Design Paradigms for Flexible Filter Accelerators

Figure 3.9: CDFG merging illustrated for three constant integer multipliers (19, 41 and 10).

3.2.3 Fine Grained Reconfigurable Fabric with Lookup Table based Constant Multipliers

Figure 3.10: 16x16 KCM Multiplier.

The Velcro and the reconfigurable datapath approaches offer only a limited degree of reconfigurability. An implementation of the required filtering operations on a fine grained reconfigurable fabric that allows bit level manipulations, offers a much higher degree

3.2 Design Paradigms for Flexible Filter Accelerators

of flexibility. The atomic unit of a fine grained fabric, typically found in a commercial FPGA, consists of a lookup table (LUT), a flip flop, and the associated interconnect logic that connects the LUTs [114]. The increased flexibility however comes at the cost of reduced efficiency. A datapath operator mapped to a fine grained fabric suffers from large area, power and delay penalties due to the interconnect network, when compared to an equivalent implementation using coarse grained datapath operators. A high degree of redundancy is built into the interconnect network in order to map any arbitrary logic onto the fabric. The configuration of both the LUTs and the interconnects is achieved through a SRAM based configuration memory [114].

Mapping a generic multiplier to the fine grained fabric incurs a large area penalty. A more efficient approach that exploits the abundance of LUTs in the reconfigurable fabric is the constant (K) coefficient multipliers (KCM) multiplier [115]. The KCM multiplier is a constant coefficient multiplier in which the partial products are generated using lookup tables and summed up in an adder tree. Consider the case of a $m \times m$ bit multiplier. A lookup table for a m bit coefficient has to generate 2^m possible output words of size $2m$. Even for medium sized coefficient wordlengths, the size of the required lookup table can be prohibitively large. A more efficient strategy would be to split the coefficient into 4 bit wide blocks, and exploit the 4 input LUTs that are commonly used in fine grained reconfigurable fabrics to generate the partial products [115].

Fig. 3.10 shows a 16x16 KCM multiplier. For each 4 bit block of the coefficient, 20 4-input LUTs are required to generate the 20 bit partial product when each block is multiplied by the 16 bit input. A total of 80 LUTs are required to generate all the partial products. Reconfiguring the filter tap for a different coefficient merely involves updating the above partial product generating LUTs. However the ability to reconfigure only a portion of a larger design at runtime, also known as dynamic partial reconfiguration is supported only in a few commercially available FPGAs [116, 117]. These FPGAs usually do not provide a fine grained access to the configuration memory. In the Virtex 4 FPGA for instance, the atomic unit of reconfiguration is a ‘frame’ of size 164 bytes [117]. A single frame configures a small portion of a column of 16 configurable logic blocks (CLBs). The configuration bits of a single CLB span across 22 frames. Hence the smallest coherent unit of reconfiguration is a set of 22 frames which completely configures a column of 16 CLBs. Each CLB has 8 LUTs. Hence 10 CLBs are required to implement the partial product generation block of a single 16x16

3.2 Design Paradigms for Flexible Filter Accelerators

KCM multiplier. The number of 16 CLB columns spanned by K KCM multipliers can be given by $\lceil (k \times 10) / 16 \rceil$ and the total size of the partial reconfiguration bitstream required to reconfigure the K KCM multipliers can be given by $\lceil (k \times 10) / 16 \rceil \times 22 \times 164$ bytes. When K is large, the average size of the partial reconfiguration bitstream per filter tap can be given by 2255 bytes. The theoretical maximum throughput of the internal configuration access port (ICAP) interface that is used to access the internal configuration memory of the Virtex 4 FPGA is 400MB/s [117]. A custom intellectual property (IP) capable of achieving the theoretically maximum partial reconfiguration throughput through the ICAP interface was recently reported [118]. The above IP uses a processor only for initialization and performs the subsequent transfer of configuration data autonomously. However practical filters required for channelization require a large number of configurable filter taps. For instance a single rate filter structure that can perform channel adaptation for WCDMA and CDMA standards requires about 858 taps [119]. Hence even with the use of dynamic partial reconfiguration, a reconfiguration latency of several milliseconds can be expected, which prohibits its applicability to applications like the cognitive radio, which switch between multiple modes at a very high rate.

3.2.4 Filter Coprocessor

Filter coprocessor approaches exploit the regularity of the FIR filter DFG prior to the application of any constant-coefficient behavioral optimization, to fold the DFGs of multiple modes onto a fixed folding set. A filter coprocessor performs the required computations in a temporal fashion, by time sharing a fixed set of generic MAC units [98-102]. Time sharing results in increased resource usage within a mode and allows variable filter lengths to be folded onto the same underlying hardware. Hence it has a considerably lower datapath area than the Velcro approach. The regularity also results in a very low control overhead. The control logic consists of simple counters to generate the coefficient and data addresses. The use of generic MAC units also allows it to support arbitrary coefficient sets. However the coprocessor approach has several drawbacks that need to be taken into account. Specialized decimation filter stages like CIC filter [65] or classical interpolation polynomial based Farrow structures [74] have either unity coefficients or small integral coefficients. Therefore using generic MAC units for the coefficient multiplications in these stages is superfluous. The inability of the coprocessor

3.2 Design Paradigms for Flexible Filter Accelerators

approach to make use of constant-coefficient based behavioral optimizations to reduce the arithmetic complexity also results in increased dynamic power consumption, when compared to the spatial style implementation. Time shared implementations also result in loss of input correlation compared to parallel implementations, further increasing the dynamic power consumption due to increased switching activity [124].

The use of generic MAC units in a time shared structure also requires the filter coefficients to be stored in a coefficient register file or RAM. Due to the limited I/O bandwidth of the accelerator core as well as the limited memory bandwidth of the processor, the high level processor routines for parameterization may require a large number of cycles to completely transfer the configuration context to the core, especially when the number of filter taps is very high. In [123] for instance, the coefficient registers are connected in a serial scan chain, and the coefficients are loaded serially on the clock edge. For 16 bit coefficients, the size of the reconfiguration data per filter tap is also 16 bits.

3.2.5 Reconfigurable Constant Multiplier based Programmable Filter

When the filter coefficients are fixed, constant multipliers can be used to reduce the multiply operations to a set of shift and add operations. Hardwired shifts are essentially free in hardware, since they only involve rewiring of the data bus lines between operators. The arithmetic complexity in constant multipliers is only dependent on the number of adders which in turn is a function of the number of non-zero digits in the coefficient. The adder count can be further reduced by coefficient optimization strategies like the canonical signed digit (CSD) that recode the coefficients to minimize the number of non-zero digits [125]. A CSD based constant multiplier has a significantly lower complexity than a generic multiplier. Hence a lot of works have attempted to introduce reconfigurability into a constant multiplier, to make it as flexible as a generic multiplier, while having a lower complexity [105-108]. All filter coefficients in a FIR filter might not have uniform precision requirements and only a few coefficients may require a high number of non-zero terms in a CSD representation of the filter coefficients. The filters in [128, 129] are inflexible in that the number of non-zero digits assigned to each tap is fixed. In [127], the atomic building unit is a p -tap, which comprises of two programmable non-zero digits as shown in Fig. 3.11. Using a bypassable delay element, multiple p -tap may be cascaded together to form a high precision coefficient.

3.2 Design Paradigms for Flexible Filter Accelerators

Hence there is greater flexibility in the allocation of non-zero digits. A p -tap can be represented by the following equation:

$$C = d_0 2^{p_0} + d_1 2^{p_1} \quad (3.12)$$

Figure 3.11: p -tap (pipeline registers not shown).

For a K bit coefficient, the permissible shift values for both p_0 and p_1 belong to the set $\{0, 1, 2 \dots K - 1\}$ while d_0 and d_1 belong to the set $\{1, -1\}$. Including the 1 bit control signal for the bypassable delay element, a $(2\lceil \log_2 (K - 1) \rceil + 3)$ bit control word is necessary to parameterize each p -tap. The average number of non-zero terms per CSD coefficient is a function of the stopband attenuation [125]. For an attenuation of about -80 dB, approximately 4 non-zero terms are required, which needs 2 p -taps. Hence a filter with 16 bit coefficients would require 22 bits of control data per filter tap for parametrization.

When multiple constant multipliers share a common input, there might be many subexpressions that are evaluated in multiple constant multipliers. Common subexpression elimination (CSE) algorithms exploit this phenomenon to further reduce the arithmetic complexity, by removing the redundant subexpressions [23]. A reconfigurable multiplier block structure is reported in [131] that makes use of the ability of the binary common

subexpression elimination (BCSE) algorithm [130], to identify more common subexpressions than when compared to a CSD based CSE algorithms in higher order filters. In the above structure the partial products of the input with all possible 3 bit subexpressions which require at least one adder, are generated in common and shared by all the filter taps, thus amortizing their cost. The binary coefficients are recoded using the BCSE algorithm to eliminate redundant subexpressions. The recoded coefficients are used to select the appropriate partial products which are shifted by their weights and finally summed up in an adder tree, as shown in Fig. 3.12. As a result of the coefficient recoding, some of the adder tree inputs can be expected to be zero. This allows some of the adders in the tree to be bypassed and prevents redundant switching. The authors report that the each 16 bit filter tap requires a 36 bit word for controlling the partial product selecting multiplexers, the programmable shifters and the steering logic required to bypass adders in the adder tree. Hence, it incurs a higher reconfiguration latency penalty than the filter coprocessor based solution.

Figure 3.12: BCSE based Reconfigurable Filter Tap.

3.3 Summary

In this chapter, we analyzed the area, power and hardware reconfiguration latency overheads that constitute the flexibility penalty of a flexible accelerator. Flexibility precludes the use of many behavioral optimizations, which make a single mode accelerator area and power efficient. Hence the additional area and power penalties need to be taken into account, while designing accelerators for implementing the computationally intensive kernels in a multistandard SDR handset. In addition to the above overheads,

3.3 Summary

the hardware reconfiguration latency is also an important consideration in applications like CRs which need to switch between different communicating channels at a much higher rate than traditional multistandard radios. The reconfiguration latency is a function of the amount of the reconfiguration data needed to parameterize the accelerator. All the components of the flexibility penalty are a function of the granularity of hardware reuse across multiple modes. The reconfiguration data and available flexibility in traditional flexible filter accelerators have been summarized in the Table 3.1.

Table 3.1: Reconfiguration data and flexibility (Number of modes supported is assumed to be N , Coefficient wordlength is assumed to be 16 bit)

Architecture	Reconfiguration Data	Flexibility	Pros	Cons
Velcro Approach [6]	$\lceil \log_2 N \rceil$	Limited Limited	Exploits low level behavioral optimizations	Large area
Coarse Grained Reconfigurable Datapath [111]	$\lceil \log_2 N \rceil$	Limited	Exploits low level behavioral optimizations	High steering logic overhead with increased standards
KCM Multipliers [115]	2255 bytes/tap	Flexible	Exploits the LUTs in FPGAs	High switching activity in interconnects
Filter Coprocessor [122]	16 bits/tap	Flexible	No coefficient preprocessing	High switching activity due to loss of input correlation
Reconfigurable Constant Multipliers-CSD [127]	22 bits/tap	Flexible	Lower power than a filter coprocessor	Needs coefficient preprocessing
Reconfigurable Constant Multipliers-BCSE [131]	36 bits/tap	Flexible	Lower power than a filter coprocessor	Needs coefficient preprocessing

The Velcro approach does not reuse any hardware across multiple modes, while the coarse grained reconfigurable datapath reuses word-level datapath operators. Since they offer only the limited flexibility to switch between a fixed set of modes, the reconfiguration overheads of the above two methods are dependent only on the number of modes that need to be supported and are independent of the filter length or the coefficient wordlength. The reconfiguration overheads for all the other architectures is a function of both coefficient wordlength and filter length. The reconfigurable constant multipliers have a higher amount of reconfiguration data than the filter coprocessor, due to the specific format in which the coefficients are stored and additional control signals required to bypass redundant adders. Among the architectures studied in this chapter,

3.3 Summary

KCM multipliers have the highest amount of reconfiguration data due to the generic nature of the underlying fine grained fabric and the large amount of data needed to reconfigure the LUTs. All the flexible filter accelerators studied in this chapter are limited to reusing fine-grained operators like LUTs or coarse grained datapath operators like adders, multipliers, MAC units and registers. Identifying reusable hardware blocks at coarser levels of granularity is essential for further reducing all the components of the flexibility penalty, beyond what is possible with traditional flexible filter accelerator structures.

Design of Multistandard Channelization Accelerators for Software Defined Radio Handsets

4.1 Introduction

The granularity of hardware reuse across different modes strongly affects the area, power and latency penalties associated with flexibility. While intellectual property (IP) level reuse has been a very important issue in system on chip (SoC) designs [132, 133], the interest in reusability had been primarily motivated by the need to reduce the design and verification time, and not the runtime reconfiguration time, area or power. A dynamically parameterizable hard IP [133], that can be used across multiple modes, also incurs additional area and power penalties due to the extra steering logic, memories and configuration registers. Reducing the above overheads and the amount of reconfiguration data requires a systematic identification of redundancies across multiple modes at the earliest stage in the design cycle. This is important for identifying which portions of the architecture need to be reprogrammable and which portions can be hardwired and reused across different modes, with little or no reconfiguration overhead. Reusing hardware at a very coarse level of granularity across multiple modes can reduce the amount of reconfiguration data, while allowing the use of high level synthesis behavioral optimizations, which reduce the area and the power.

The functionally different channelization tasks of channel selection, interferer attenua-

tion, pulse shaping and sample rate conversion can be reduced to a set of filtering and decimation operations [62]. The Velcro approach [134] and the coprocessor approach [121] represent two ends of the traditional multistandard filter accelerator design space. The Velcro accelerator consists of multiple single-mode accelerators stacked in parallel, each of which have been optimized for a specific standard. While supporting multiple standards in this fashion incurs a significant area penalty, the Velcro accelerator has the advantage of having a dynamic power consumption which is comparable to that of a single mode accelerator. However its flexibility is limited to the ability to switch between a known set of standards, and cannot support in-field upgradeability. The coprocessor approach maps the filtering function onto a set of generic multiply and accumulate (MAC) units and is capable of accommodating arbitrary filter specifications. However its flexibility comes at the expense of a significant dynamic power consumption penalty when compared to a single mode accelerator. Hence, both the above accelerator paradigms falls short of the flexibility and power efficiency requirement of a SDR handset. This chapter tackles the problem of evolving an alternate channelization accelerator paradigm that has lower dynamic power consumption than the coprocessor approach, while still being sufficiently flexible to switch between a known set of standards or support a new standard by means of an in-field upgrade. The proposed architecture achieves these goals by reducing the computational load assigned to generic MAC units and identifying opportunities for reusing hardwired structures at a very coarse level of granularity across multiple standards.

The filtering and decimation signal processing tasks can be efficiently performed by a multistage decimation filter structure. The filter stages in the above structure represent a naturally coarse level of granularity for exploring hardware reuse. However the ability to hardwire and reuse these filter stages is limited by their dependency on standard specific channelization parameters, which necessitate the use of parameterizable flexible filter structures like the ones studied in chapter 3. To the best of our knowledge, the problem of systematically hardwiring and reusing stages of a multistage decimation filter across an arbitrary number of modes has not been addressed in the literature. Prior works which have focused on the problem of reusing filter stages across multiple modes, have been largely limited to a few modes, and are not easily extendible to cases where an arbitrary number of modes need to be supported [81]. Besides the reductions in the number of multiplications per second offered by a multistage decimation filter over a

4.2 Proposed Multistandard Channelization Accelerator Design Flow

single stage filter, additional complexity reductions can be achieved by using specialized filter structures to implement the stages like cascaded integrator comb (CIC) filters [65], halfband filters [56], and Farrow's structures [73]. These stages have significantly lower implementation cost when compared to generic FIR decimation stages, but are effective only at certain stages in the decimation chain, or for certain decimation factors. Hence, the partitioning of the decimation load into multiple stages needs to take multiple factors into account. Firstly, it has to maximize the use of specialized filter structures, when possible, to reduce the arithmetic complexity. Secondly the partitioning strategy needs to handle arbitrary rational decimation factors, and should be capable of finding opportunities for hardwiring and reusing decimation stages across multiple standards.

4.2 Proposed Multistandard Channelization Accelerator Design Flow

4.2.1 A Mathematical Formulation of the Multistandard Channelization Problem

Assume that ' P ' wireless standards have to be supported by the channelization accelerator. Let the C/N requirement of the j th standard be C_j . Let the SRC factor of the j th standard be M_j . The $\Sigma\Delta$ ADC design determines the OSR, M_j , which is necessary to achieve the desired dynamic range. Note that in typical $\Sigma\Delta$ ADC decimation filter designs, the OSR parameter is expressed with respect to the Nyquist rate. In this thesis, the OSR at any position of the channelization accelerator is always expressed with respect to the symbol rate. Assume that M_j can be factorized into n_j factors, so that the required decimation is performed by n_j filtering stages. If the final symbol rate (or chip rate for spread spectrum systems) is f^j , the ADC sampling rate can be expressed in terms of the decimation factors as follows [54]:

$$M_j f^j = \left(m_1^j \cdot m_2^j \cdot m_2^j \dots m_{n_j}^j \right) f^j. \quad (4.1)$$

When the pulse shaping for a particular mode is performed by a RC transmit filter or a pair of matched RRC transmit and receive filters, the bandwidth that has to be protected from aliasing is a function of the symbol rate f^j and the roll-off factor α_j , as follows:

$$f_{pb}^j = \frac{f^j}{2} (1 - \alpha_j) \quad (4.2)$$

4.2 Proposed Multistandard Channelization Accelerator Design Flow

Figure 4.1: Position of alias intervals.

The term f_{pb}^j corresponds to the passband edge, as shown in Fig. 4.1. The roll-off factor, α_j , lies in the interval $[0,1]$ [54]. For most commonly used standards, which use Gaussian filters for transmit side pulse shaping, the BT product of the Gaussian filter lies in the interval, $(0, 0.5)$ [135]. If the j th standard used a Gaussian filter with a BT product, β_j , the band edge of the information bandwidth that needs to be protected from aliasing can be given by Eq. 4.3.

$$f_{pb}^j = f^j \beta_j \quad (4.3)$$

Assuming that the desired channel sensitivity level is given by S_j dBm, each decimation stage has to ensure that any potential interferer, blocker, or quantization noise that can alias into the passband interval $(-f_{pb}^j, f_{pb}^j)$ has to be attenuated to the level $(S_j - C_j)$ dBm. This criterion is henceforth referred to as the ‘anti-aliasing criterion’ in the current work. Consider the design of the i th decimation stage of the j th standard, which decimates by the factor m_i^j . The input and output sampling rates of the above stage can be given by Eq. 4.4 and Eq. 4.5 respectively.

$$f_{in_i}^j = \left(\prod_{k=i}^{n_j} m_k^j \right) f^j \quad (4.4)$$

$$f_{out_i}^j = \left(\prod_{k=i+1}^{n_j} m_k^j \right) f^j \quad (4.5)$$

The potential alias components for this stage lie in the frequency intervals of bandwidth $2f_{pb}^j$, centered around integral multiples of output sample rate $f_{out_i}^j$, as shown in Fig. 4.1.

The potential aliasing components within these bands may comprise of interferers, blockers or just quantization noise from the $\Sigma\Delta$ ADC. The degree of stopband attenuation required to suppress the aliasing components to the noise floor, depends

4.2 Proposed Multistandard Channelization Accelerator Design Flow

on the interference levels within the aliasing bands. Designing a decimation filter to strictly conform to the above constraints requires a multi-stopband filter with multiple stopband attenuation values, and multiple don't-care regions. The unattenuated don't-care regions, do not cause aliasing into the passband for the current stage, and can be removed by the latter decimation stages [61]. The design methodologies for low pass FIR filters being more mature, with easily available design tools, it is more common to implement the decimation filters using low pass FIR filters, which provide uniform stopband attenuation over both the aliasing and don't-care regions [61]. The required stopband attenuation T_i^j can be chosen to satisfy the worst case attenuation requirements, among all the aliasing bands. The requirement of preventing aliasing in the passband interval, $(-f_{pb}^j, f_{pb}^j)$ can be achieved by attenuating all the components above $(f_{out_i}^j - f_{pb}^j)$, by T_i^j . The frequency, $(f_{out_i}^j - f_{pb}^j)$, represents the starting boundary of the first aliasing interval. Hence, the passband edge, fp_i^j , and the stopband edge, fs_i^j of the i th decimation stage can be given as follows:

$$fp_i^j = f_{pb}^j \quad (4.6)$$

$$fs_i^j = f_{out_i}^j - f_{pb}^j = \left(\prod_{k=i+1}^{n_j} m_k^j \right) f^j - f_{pb}^j \quad (4.7)$$

When RC or RRC filters are used for pulse shaping, the passband edge, wp_i^j , and the stopband edge, ws_i^j , of the i th decimation stage in radians (normalized by 2π) can be derived as shown below:

$$wp_i^j = \left(\frac{1 - \alpha_j}{2 \prod_{k=i}^{n_j} m_k^j} \right) \quad (4.8)$$

$$ws_i^j = \left(\frac{\left[2 \prod_{k=i+1}^{n_j} m_k^j \right] - 1 + \alpha_j}{2 \prod_{k=i}^{n_j} m_k^j} \right) \quad (4.9)$$

4.2 Proposed Multistandard Channelization Accelerator Design Flow

When the pulse shaping is performed by a Gaussian filter, the corresponding band edges can be given by Eq. 4.10 and Eq. 4.11, as shown below:

$$wp_i^j = \left(\frac{\beta_j}{n_j \prod_{k=i}^j m_k^j} \right) \quad (4.10)$$

$$ws_i^j = \left(\frac{\left[\prod_{k=i+1}^j m_k^j \right] - \beta_j}{n_j \prod_{k=i}^j m_k^j} \right) \quad (4.11)$$

The expressions for stopband and passband edges highlight the main sources of variability in a multistandard multistage decimation filter design. It can be seen that stopband edge, passband edge and stopband attenuation parameters of each stage in the multistandard multistage decimation filter are dependent on the standard specific parameters. This limits the possibilities for hardwiring a filtering stage and directly reusing it across multiple standards without any changes. The need for reprogrammability precludes the use of constant coefficient filter optimizations and necessitates the use of generic multipliers.

4.2.2 Eliminating the Dependency of Filter Stages on Standard Specific Parameters

It was shown in Eq. 4.8 - 4.11 that the dependency of the filter stages on the standard specific parameters prevents them from being readily reused across multiple standards. This section discusses some specific conditions under which the filter specifications can be modified to eliminate this dependency, without changing their functionality. Consider the design of the k th decimation stage in a decimation filter chain for the j th standard. Assume that this stage decimates by the factor p . An ideal decimation filter for this stage has a brickwall response, cutting off all the frequency components above the frequency π/p , as shown in Fig. 4.2. However the ideal brickwall characteristics requires an infinitely long filter and hence is not feasible for practical implementation. A more practical filter exploits the fact that the band containing the useful information,

4.2 Proposed Multistandard Channelization Accelerator Design Flow

fp_k^j is smaller than π/p . This allows the frequency domain requirement to be relaxed by introducing a transition band, which prevents aliasing only in the interval $(-fp_k^j, fp_k^j)$. This is illustrated in Fig. 4.3. As shown in the previous section, the band edges and stop band attenuation introduced for practical implementation depend on standard dependent parameters, which limit their reusability.

Figure 4.2: Ideal decimation filter stage for factor ‘p’.

Figure 4.3: Feasible decimation filter stage for factor ‘p’.

Consider the case of two multistage decimation filters corresponding to two different standards, henceforth referred to as standard-1 and standard-2. Assume that the first standard performs pulse shaping through a pair of matched transmit and receiver RRC filters, while the second standard performs the same using a Gaussian transmit filter. Let $H_1(z)$ be one of the decimation stages in the first filter, and $H_2(z)$ be one of the decimation stages in the second filter. The respective filter specifications of these two stages are dependent on the input OSR, decimation factor, attenuation requirement and the pulse shaping parameters. Now consider a special case where both $H_1(z)$ and $H_2(z)$ lie at the same OSR pq with respect to their respective symbol rates and both decimate by the same factor p , as indicated in Fig. 4.4. The band edges and stopband attenuation of both the stages have to be specified such that the potential alias components are suppressed to the noise floor and their respective passband intervals

4.2 Proposed Multistandard Channelization Accelerator Design Flow

Figure 4.4: Position of $H_1(z)$ and $H_2(z)$ in the decimation filter chain.

are free from aliasing. Let the roll off factor for the first standard be α_1 and the BT product for the second standard be β_2 . The passband intervals of the two standards (frequency values normalized by 2π) that have to be protected from aliasing, can be given by $\left(-\frac{1-\alpha_1}{2pq}, \frac{1-\alpha_1}{2pq}\right)$ and $\left(-\frac{\beta_2}{pq}, \frac{\beta_2}{pq}\right)$ respectively. For a filter which decimates by a factor of p , the potential alias images lie around integral multiples of $\frac{1}{p}$ (frequency value normalized by 2π). The starting boundary of the first aliasing intervals of $H_1(z)$ and $H_2(z)$ lie at $\left[\frac{1}{p} - \frac{1-\alpha_1}{2pq}\right]$ and $\left[\frac{1}{p} - \frac{\beta_2}{pq}\right]$ respectively. Let the required stopband attenuation for $H_1(z)$ and $H_2(z)$ be A_1 and A_2 respectively. The anti-aliasing criterion for $H_1(z)$ and $H_2(z)$ can be fulfilled by placing the passband edge at boundary of the band containing the useful information in standard-1 and standard-2 respectively, and by placing the stopband edges at the starting boundary of the first alias intervals, while attenuating by A_1 and A_2 in their respective stopbands. The decimation filter responses, $H_1(z)$ and $H_2(z)$ are shown in Fig. 4.5(a) and Fig. 4.5(b) respectively. It can be seen that the filter specifications show a dependency on the standard dependent pulse shaping and stopband attenuation parameters. In a multistage decimation filter, the overall band edge is determined by the last stage filter. The stages prior to the last stage merely perform a decimation operation while ensuring that the passband is free from aliasing, and do not have an explicit band-edge shaping role. Hence the band-edges and attenuation specification of all the stages prior to the last stage, can potentially be modified without affecting the functionality of the overall multistage filter, as long as the modified specifications for the individual stages satisfy the anti-aliasing criterion in the passband.

Consider a modified filter stage, $H_{12}(z)$, whose stopband attenuation is fixed at $\max(A_1, A_2)$ and band edges are designed for the boundary conditions of the pulse shaping parameters ($\alpha = 0, \beta = 0.5$). These boundary conditions correspond to the maximum permissible passband intervals, and hence $H_{12}(z)$ protects a wider band from aliasing, than either $H_1(z)$ or $H_2(z)$. Fig. 4.5(c) shows the modified filter stage $H_{12}(z)$ with

4.2 Proposed Multistandard Channelization Accelerator Design Flow

the passband and stopband edges fixed at $\frac{1}{2pq}$ and $\left(\frac{1}{p} - \frac{1}{2pq}\right)$ respectively. It can be seen that, the modified specifications satisfy the anti-aliasing criterion for both the standards. Any unattenuated, undesired component will be removed by the succeeding decimation filter stages. It can be inferred from these observations that the presence of filtering stages which decimate by the same factor, at the same OSR in different multistage decimation filters, allows them to be replaced with a modified specification which can be hardwired and reused across the multiple standards without any changes. However this band-edge modification strategy cannot be used for the last stage filter, since it determines the overall band-edge of the multistage decimation filter.

Figure 4.5: Frequency responses of $H_1(z)$, $H_2(z)$ and $H_{12}(z)$.

4.2.3 Fixed Factorization Method for Multistandard Channelization

The multistandard channelization accelerator design flow proposed in this section differs from the existing filter accelerator design paradigms in the fact that, the commonalities in multiple channelization specifications are identified at the earliest possible stage in the design cycle. This allows a unified high level and architectural synthesis to be performed for mapping multiple modes onto the same underlying hardware. It also allows a significant portion of the accelerator to be hardwired and reused across multiple modes, with minimal overheads for reconfiguration and parametrization. The

4.2 Proposed Multistandard Channelization Accelerator Design Flow

hardwired portions of the accelerator are also able to make use of various constant-coefficient filter optimizations for reducing the area and power.

In the arbitrary factorization of the decimation ratio in Eq. 4.1, the number of decimation factors and the value of the decimation factors themselves can show considerable variation across standards. This section introduces an alternate factorization strategy that exploits the observations in the previous section, and maximizes the number of decimation filter stages that lie at the same OSR and decimate by the same factor, across multiple standards. Consider a factorization of a rational decimation factor, M_j as shown in Eq. 4.12.

$$M_j = k_j \cdot m_1 \cdot m_2 \cdot m_3 \cdot m_4 \dots m_n \quad (4.12)$$

where k_j represents a standard dependent rational factor, while the factors m_i , where $1 \leq i \leq n$, represent a set of n integral factors, that are common to all standards. The order of the factors in Eq. 4.12, also represents the actual order in which the multistage decimations are performed. Allowing the first factor, k_j , to be rational allows any arbitrary rational factor M_j to be factorized into the above form. Assume that each of the P decimation factors corresponding to the P supported standards, have been factorized in the above form. Consider the case of the decimation stages which implement the decimation factor m_i , where $1 \leq i \leq n-1$. In general the band-edges and alias attenuation of this stage, in each of the P standards, are dependent on standard specific parameters and hence, cannot be readily reused across multiple standards. However it can be observed that in all the P standards, this filter stage lies at the same input OSR, $\prod_{k=i}^n m_k$, and decimates by the same factor, m_i . This implies that the standard-dependent filter stages for the factor m_i , can be replaced by a modified filter, which satisfies anti-aliasing criterion in the passband for all the standards. Hence the filter stages implementing the factors m_1 to m_{n-1} , can be hardwired and reused across all the supported modes without any changes. Only the last decimation stage which decimates by the factor m_n has to be programmable to support variable band-edge specifications.

The factorization in Eq. 4.12 also requires decimation by a rational factor k_j , prior to the integral decimations. Assume that the rational decimation factor can be further decomposed into an integral factor k_{int}^j and a fractional decimation factor k_{fra}^j as shown

4.2 Proposed Multistandard Channelization Accelerator Design Flow

below:

$$k^j = k_{int}^j \cdot k_{fra}^j \quad (4.13)$$

k_{fra}^j is assumed to lie in the interval (1, 2). For integral decimation by k_{int}^j in the first stage, cascaded integrator comb (CIC) filters are very efficient due to the absence of any multipliers operating at the ADC sampling rate [65]. Fractional decimation by the factor k_{fra}^j can be performed by a transpose Farrow structure based on classical interpolating polynomials, like Lagrange or B-splines [74]. Both structures are advantageous from the perspective of supporting multiple standards, because of the ease of handling variable decimation factors and the absence of an explicit, standard dependent band-edge. The zeroes of both these structures lie at the center of the potential aliasing bands. The worst case alias attenuation in a CIC filter is determined by its order, while that of the transpose Farrow structure depends on the underlying time domain polynomial which is used to approximate sinc response of an ideal reconstruction filter [74]. Both the CIC order as well as the underlying polynomial of the transpose Farrow structure can be chosen to satisfy the worst case attenuation requirement among all the P standards.

In a classical recursive CIC structure, changing the sample rate merely involves changing the downsampling clock in between the integrator and comb stages [65]. However the recursive CIC structure has large wordlength adders in the integrators, which operate at the input oversampled rate, which results in high power consumption [66]. When the CIC filter decimation factor R_d is of the form $R_d = 2^s$, the transfer function of a N_c th order CIC filter can be factorized as follows:

$$H_{cic}(z) = \prod_{i=0}^{s-1} \left(1 + z^{-2^i}\right)^K \quad (4.14)$$

The Noble identities can be used for commuting the subsequent downsampling operation with the filter stages for reducing the computational rate, resulting in a structure called as the non-recursive CIC filter (NRCIC) [67]. The resulting structure has s stages. Let the highest possible decimation factor of the NRCIC filter be $2^{s_{max}}$. Assuming that the stages have been hardwired for the worst case CIC order ' K ', variable sample rates can be supported by inserting additional steering logic to bypass unwanted stages, as shown in Fig. 4.6.

4.2 Proposed Multistandard Channelization Accelerator Design Flow

Figure 4.6: Programmable nonrecursive CIC structure.

A Farrow structure is very efficient for fractional interpolation. It reduces the general requirement of a time-varying reconstruction filter, with a structure that consists largely of linear time-invariant subfilters. The only time-varying components are the so called μ -multipliers, which share a common input called then fractional sample interval. The transpose Farrow structure, can be obtained by a transposition of the traditional Farrow structure [136]. The zeroes of the resulting transfer function lie at the integral multiples of the output sample rate, and hence is suitable for a fractional decimation operation. As long as the underlying interpolating polynomial is chosen to accommodate the worst case attenuation requirement, reconfiguring the transpose Farrow structure for a different fractional decimation factor merely consists of re-initializing the control word of the counter which generates the fractional interval value, as shown in Fig. 4.7 [60].

Figure 4.7: Transpose Farrow structure.

The last stage filter, which decimates by the factor m_n for all the modes, determines the overall standard-dependent band edge. Depending on the division of the pulse shaping load between the transmit and receive side, this filter might need to perform a pulse

4.2 Proposed Multistandard Channelization Accelerator Design Flow

shaping function. When no pulse shaping is performed by the last stage, the stopband edge can be placed at the boundary of the channel bandwidth, while the passband edge is still determined by the transmit side pulse shaping filter. For RC pulse shaping at the transmit side, the passband edge, wp_n^j and stopband edge, ws_n^j , in radians (normalized by 2π) can be given as follows:

$$wp_n^j = \left(\frac{1 - \alpha_j}{2m_n} \right) \quad (4.15)$$

$$ws_n^j = \left(\frac{f_c^j}{4m_n f^j} \right) \quad (4.16)$$

where f_c^j represents the inter-channel spacing for the j th standard. For Gaussian pulse shaping, the stopband edge is identical to Eq. 4.16, while the passband edge can be given as:

$$wp_n^j = \left(\frac{\beta_j}{m_n} \right) \quad (4.17)$$

When the last stage filter implements RRC pulse shaping, the passband edge is identical to Eq. 4.15, while the stopband edge is given by:

$$ws_n^j = \left(\frac{1 + \alpha_j}{2m_n} \right) \quad (4.18)$$

A typical multistage decimation filter capable of multistandard support, needs the individual stages to be reprogrammable to accommodate variable standard specific parameters like the band-edge specification and the alias attenuation requirement. The proposed approach alleviates this problem, allowing all the stages prior to the last stage to be hardwired and reused across multiple modes, with very little reconfiguration overheads. For these stages, switching between different standards, only requires the control words for the NRCIC filter in the first stage and the fractional sample interval generator of the transpose Farrow structure in the second stage to be parameterized. This allows the hardwired stages to be implemented at a very low area and power cost, by making use of constant coefficient filter optimizations. Only the last stage filter needs to be completely programmable and needs the use of generic multipliers.

4.3 Practical Filter Design Considerations

Two crucial design issues have to be still addressed before the proposed factorization strategy can be used in a practical implementation: the optimum factorization of an arbitrary rational sample rate conversion factor in the form required by Eq. 4.12 and the identification of the ideal filtering structure for implementing the decimation stages.

4.3.1 Design Considerations for NRCIC Stage

While the NRCIC structure has the advantages of low complexity and low reconfiguration overhead, it suffers from a non uniform passband. The magnitude response of a K th order CIC filter, which decimates by a factor of R is given as follows [65]:

$$H_{cic}(\omega) = \frac{1}{R^K} \left[\frac{\sin(\omega R/2)}{\sin(\omega/2)} \right]^K \quad (4.19)$$

For low frequencies, $H_{cic}(\omega)$ can be approximated by $[\text{sinc}(\omega R/2)]^K$. The passband droop can be compensated by cascading a droop correction filter $C(z^{-1})$ as shown in Fig. 4.8(a). Using Noble identities, the decimation factor can be commuted to the right of the $C(z^{-1})$, as shown in Fig. 4.8(b). The compensated CIC transfer function is given by $H_{cic}(z^{-1})C(z^{-R})$. If $C(z^{-1})$ has a magnitude response of $[\text{sinc}(\omega/2)]^{-K}$, within the intervals $(-Rf_{pb}^j/f^j, Rf_{pb}^j/f^j)$, the interpolated filter, $C(z^{-R})$, has a magnitude response of $[\text{sinc}(\omega R/2)]^{-K}$ within the passband interval $(-f_{pb}^j/f^j, f_{pb}^j/f^j)$, and hence can correct the CIC droop within this interval. However, the passband edge being a function of the standard dependent pulse shaping parameters, the droop correction filter needs to be programmable to support multiple standards. The proposed factorization strategy in Eq. 4.12 guarantees a minimum OSR of $\prod_{i=1}^n m_i$ at the output of the CIC stage, which can be exploited for hardwiring and reusing the droop correction filter across multiple standards, as shown below.

Figure 4.8: CIC droop correction.

4.3 Practical Filter Design Considerations

The input sample rate to the CIC stage can be expressed in terms of the symbol rate f^j , as follows:

$$f_{in}^j = k_{int}^j \cdot k_{fra}^j \cdot \prod_{i=1}^n m_i \cdot f^j \quad (4.20)$$

Preserving the integrity of the signal within the passband interval requires f^j to be greater than or equal to $2f_{pb}^j$, as per the Nyquist criterion [137]. Taking this constraint into account, Eq. 4.20 can be used to derive the following inequalities:

$$f_{in}^j \geq 2k_{int}^j \cdot k_{fra}^j \cdot \prod_{i=1}^n m_i \cdot f_{pb}^j \quad (4.21)$$

$$\frac{f_{pb}^j}{f_{in}^j} \leq \frac{1}{2k_{int}^j \cdot k_{fra}^j \cdot \prod_{i=1}^n m_i} \quad (4.22)$$

where f_{pb}^j/f_{in}^j represents the passband edge of the the j th standard, in radians (normalized by 2π). Since the fractional decimation load k_{fra}^j lies in the interval (1, 2), Eq. 4.22 can be reformulated as follows:

$$\frac{f_{pb}^j}{f_{in}^j} \leq \frac{1}{2k_{int}^j \cdot \prod_{i=1}^n m_i} \quad (4.23)$$

Hence if the droop correction filter is designed to have a passband edge of $1/(2 \prod_{i=1}^n m_i)$, the interpolated response $C(z^{-k_{int}^j})$ compensates the CIC response for all frequencies below $1/(2k_{int}^j \cdot \prod_{i=1}^n m_i)$. Such a filter satisfies the droop correction requirement over a band wider than the passband of any standard, as indicated by Eq. 4.22, and hence can be hardwired and reused by all standards.

4.3.2 Design Considerations for Fractional Decimation

The proposed factorization in Eq. 4.12 requires a fractional decimation stage to be present in the decimation chain to factorize arbitrary rational factors into the required regular form. The transpose Farrow structure can be used for performing the fractional decimation operation, as it attenuates the potential alias components [136]. Transpose

4.3 Practical Filter Design Considerations

Farrow structures based on classical interpolating polynomials like Lagrange or B-splines, have the major drawback of having a poor wideband response. Hence they may not be able to sufficiently attenuate potential alias components, if the fractional decimation is performed in a low OSR environment. The fixed factorization design strategy requires the fixed decimation stages to be designed for the maximum alias attenuation requirement among all supported standards.

An ideal reconstruction filter has a brickwall frequency response which removes all the spectral images in the input. The corresponding time domain response is a sinc function, which has an infinite temporal support. The Farrow structure uses piecewise polynomials with a finite temporal support to approximate the ideal sinc response. Fig. 4.9 plots the magnitude response of the transpose Farrow structure, for different underlying polynomials. The x-axis shows the frequency normalized with respect to the sampling rate at the output of the fractional decimation stage, $(\prod_{i=1}^n m_i) f^j$. The zeroes of the transpose Farrow structure lie around the integral multiples of $(\prod_{i=1}^n m_i) f^j$, which are also midpoints of the aliasing intervals. It can be seen that the alias attenuation is very high at the center of the alias intervals, and is lower near the edges of the alias intervals. The worst case attenuation for the j th standard occurs at $f_{wa}^j = (\prod_{i=1}^n m_i f^j - f_{pb}^j) / f^j (\prod_{i=1}^n m_i)$, which is the starting boundary of the first alias interval, normalized by the output sample rate, $(\prod_{i=1}^n m_i) f^j$. Since the inequality $f^j \geq 2f_{pb}^j$ has to be satisfied to prevent aliasing within the passband, a relationship between f_{wa}^j and the output OSR, $\prod_{i=1}^n m_i$, can be established as follows:

$$f_{wa}^j = 1 - \frac{f_{pb}^j}{f^j \prod_{i=1}^n m_i} \quad (4.24)$$

$$f_{wa}^j \geq 1 - \frac{1}{2 \prod_{i=1}^n m_i} \quad (4.25)$$

Reducing the output OSR, $\prod_{i=1}^n m_i$, of the fractional decimation stage shifts f_{wa}^j to the left, away from the first zero of the transpose Farrow structure, resulting in a lower alias attenuation. Increasing $(\prod_{i=1}^n m_i)$ has the inverse effect of shifting f_{wa}^j towards the first zero, thereby increasing the degree of stopband attenuation. Fig. 4.9 indicates that for a given output OSR, the alias attenuation is also a function of the specific class of the

4.3 Practical Filter Design Considerations

Figure 4.9: Frequency response of a transpose Farrow Structure.

interpolating polynomial used for designing the transpose Farrow structure. Among the classical interpolating polynomials, B-splines offer the closest approximation to the ideal sinc response and the highest alias attenuation for the same temporal support. The polynomial has to be chosen such that for all supported standards, the transpose Farrow structure offers a sufficient amount of alias attenuation for the required output OSR of $(\prod_{i=1}^n m_i) f^j$.

4.3.3 Partitioning of the Decimation Load

Partitioning the arbitrary rational factor M^j into multiple stages, needs to take the specific constraints of available decimation filter structures into account. The non recursive implementation of the CIC structure, needs the CIC decimation factor to be a power of two. For the subsequent integral decimation, halfband filters are highly efficient structures for a decimation factor of two [56]. Halfband filters have an advantage over generic FIR decimation stages, due to the fact that nearly 50% of the filter coefficients are zero. When the coefficient symmetry due to a linear phase is also taken into account, only about 25% of the filter taps need to be actually implemented. Hence they have relatively low area and power costs, when compared to generic FIR filters. Halfband filters, however have the limitation that the passband and stopband edges are constrained to be symmetrical around 0.5π radians, and hence cannot be used for stages whose band edges do not satisfy the above constraint. Performing a bulk of the decimation load using the NRCIC filter and halfband filters requires the decimation factors of two or powers of two to be extracted from M^j . This can be achieved by

4.4 Experimental Synthesis Results

splitting M^j into fractional and integral factors as follows:

$$M^j = 2^{x_j} \cdot \left(\frac{M^j}{2^{x_j}} \right) \quad (4.26)$$

where $x_j = \lfloor \log_2(M^j) \rfloor$. The factor, 2^{x_j} , represents the largest power of two, less than or equal to M^j . The fractional decimation factor ($M^j/2^{x_j}$) lies in the interval $[1, 2)$ and can be assigned to the transpose farrow structure. The integral factor, 2^{x_j} , needs to be further split between the the first stage NRCIC filter and the subsequent integral decimation stages after the transpose Farrow structure. Splitting the integral load is determined by the output OSR required by the chosen transpose Farrow structure for satisfying the anti-aliasing criterion for all the standards. Consider a scenario where a simple polynomial is used for implementing the transpose Farrow structure, which offers the required degree of alias attenuation only at a higher output OSR. This reduces the decimation load on the NRCIC stage while increasing the decimation load after the fractional decimation stage, necessitating increased number of hardwired filter stages after the fractional decimation. Conversely, a higher order polynomial may offer the required degree of alias attenuation even at low OSRs, which shifts the bulk of the integral decimation load to the NRCIC stage. Assuming that the output OSR required by a particular transpose Farrow structure is $2^{x_{out}}$, the decimation load of the NRCIC stage can be given by $2^{x_j - x_{out}}$. The decimation by a factor of $2^{x_{out}-1}$ after the fractional decimation stage can be performed by $(x_{out} - 1)$ halfband stages. The final decimation by factor of two can be performed by a programmable FIR filter, which provides a standard dependent band-edge, and a standard dependent pulse shaping function, if the pulse shaping load is split between the transmit and receive sides.

4.4 Experimental Synthesis Results

In this section, we demonstrate a practical implementation of a multistandard channelization accelerator, based on the proposed design paradigm, and compare its area and power penalties to that of a Velcro style accelerator and a filter coprocessor. The standards that have been considered for the current analysis are Global System for Mobile Communications (GSM), Wideband Code Division Multiple Access (W-CDMA), IEEE 802.11a, and Worldwide Interoperability for Microwave Access (WiMAX). The design of the channelization function is strongly coupled to the design of the analog front-end

4.4 Experimental Synthesis Results

design and the $\Sigma\Delta$ ADCs. We assume that a programmable zero-IF analog front-end architecture is used to downconvert the channels of interest to baseband. In line with the fixed digitization approach [20], the bandwidth of the low pass analog baseband filter is assumed to be fixed according to the widest possible channel bandwidth (10 MHz, for WiMAX and IEEE 802.11a). This low pass filter removes all possible out of band interferers. Only the interferers within this band are considered for designing the channelization accelerator. The analog-to-digital conversion is assumed to be performed by the multimode $\Sigma\Delta$ ADC, proposed in [48], which is capable of handling GSM, W-CDMA, IEEE 802.11a and WiMAX standards. The channelization and sample rate conversion specifications of the individual standard are given in Table 4.1.

Table 4.1: Channelization Parameters

Standard	Channel spacing	Pulse shaping	Symbol/chip rate	Sampling rate	Decimation factor
GSM	200 kHz	Gaussian(BT= 0.3)	270.833 KSymbol/s	32 MHz	118.154
W-CDMA	5 MHz	RRC($\alpha = 0.22$)	3.84 Mcps	64 MHz	16.667
IEEE 802.11a	20 MHz	Gaussian(BT = 0.5)	12 MSymbol/s	160 MHz	13.333
WiMAX	20 MHz	RRC($\alpha = 0.25$)	16.704 MSymbol/s	160 MHz	9.578

The decimation filter stages of each standard have to be designed such that the interferers and quantization noise, unattenuated by the previous stages, are attenuated to the standard dependent noise floor. To demonstrate the area overheads of a Velcro style multistandard accelerator, four different multistage decimation filters were implemented, corresponding to each of the four standards under consideration. A cubic B-spline based transpose Farrow structure was used for performing the fractional decimation in all the standards. This filter satisfies the alias attenuation requirement of all the modes, for an output OSR of 4. Spline interpolation belongs to a general class of interpolation, in which the signal is reconstructed using interpolation coefficients, obtained by passing the input samples through a prefilter [74]. The cubic B-spline prefilter is an unstable IIR filter, which cannot be implemented directly. A linear phase FIR approximation of the prefilter [138], has been used in the current implementation.

4.4 Experimental Synthesis Results

The decimation load in each of the single mode decimation filters is partitioned according to the factorization in Table 4.2. The integral decimation in the first stage is performed by a fifth order NRCIC filter for all the modes. The droop correction filter of the NRCIC stage and the cubic B-spline prefilter in the second stage, can be clubbed together into a single filter, whose coefficients can be obtained by a convolution of the above two filter responses. The final decimation by a factor of 4 is performed by a hardwired constant-coefficient halfband and a hardwired FIR stage, in each of the four standards. Both the above filter stages decimate by a factor of 2. The last FIR decimation stage provides the standard dependent band-edge. For W-CDMA and WiMAX standards, the last stage FIR filters also perform a RRC pulse shaping function. The hardwired filter stages have been optimized by the H_{cub} multiple constant multiplication (MCM) algorithm [25]. The H_{cub} algorithm results in very low adder counts when the coefficients have a wordlength less than 32 bits and the number of coefficients are less than 100. The HDL level code of H_{cub} based constant-coefficient filters can be readily obtained through the SPIRAL online code generator [166].

Table 4.2: Factorization of SRC Factor

Standard	NRCIC	Farrow	Halfband	FIR
GSM	16	1.8461	2	2
W-CDMA	4	1.0416	2	2
IEEE 802.11a	2	1.6667	2	2
WiMAX	2	1.1972	2	2

A multistandard decimation filter was also designed using the methodology proposed in the current work, as shown in Fig. 4.10. It also uses the same factorization scheme suggested in Table 4.2, and the same cubic B-spline based transpose Farrow structure, that was used for fractional decimation in the single mode decimation filters. Hence in any particular mode, both the single mode and the proposed multistandard implementations would have the same number of stages, the main difference being that in the single mode case, all stages are hardwired, whereas in the multistandard case, some of the stages are programmable/configurable. The integral decimation in the first stage was performed by a programmable NRCIC filter. The decimation by a factor of 4 was

4.4 Experimental Synthesis Results

Figure 4.10: Proposed multistandard decimation filter.

performed by constant-coefficient hardwired halfband filter, and a programmable time-shared FIR filter, which provides the standard dependent band-edge. The band-edge modification strategy, proposed in the current work, allows the same halfband filter to be hardwired and reused by all standards without any reconfiguration overheads. The hardwired stages in the multistandard accelerator are also optimized using the H_{cub} MCM algorithm [25]. The last stage FIR filter decimates by a factor of 2, and is implemented in polyphase form with each of the two subfilters mapped onto a single MAC unit. A register file capable of storing 16 16-bit coefficients was instantiated for each subfilter, taking into account the filter length requirements in the last stage. Both the Velcro style accelerator and the proposed multistandard accelerator were implemented using a TSMC $0.18\mu\text{m}$ process. Their cell areas were estimated using the Synopsys Design Compiler. The area estimates in terms of the gate count in Table 4.3 were obtained by normalizing these cell areas by the cell area of a two input NAND gate of the same library.

The experimental synthesis data shows that for the standards under consideration, the proposed multistandard channelization accelerator approach can result in about 65% reduction in the area, when compared to a Velcro style multistandard accelerator implementation for the four standards under consideration. With increasing number of standards, the achieved reductions with respect to a Velcro style implementation can be expected to increase. The area advantage over the Velcro approach comes at the expense of increased power consumption and reconfiguration time. A Velcro style multistandard accelerator requires only a small control word to switch between multiple single mode accelerators. In the proposed approach, the last stage filter is completely programmable, while both the NRCIC and transpose Farrow structure need a control word for parametrization.

4.4 Experimental Synthesis Results

Table 4.3: Gate count of different channelization accelerator implementations

Standard	Gate count
GSM accelerator	26654
W-CDMA accelerator	23324
IEEE 802.11a accelerator	15514
WiMAX accelerator	21110
Velcro style accelerator (GSM, 802.11a, W-CDMA and WiMAX modes)	86602
Proposed multistandard accelerator (GSM, 802.11a, W-CDMA and WiMAX modes, with in-field upgradeability)	30033

The proposed structure will always be more expensive in terms of power consumption when compared to a single mode accelerator due to the additional overheads of having a programmable FIR stage and a programmable NRCIC stage. The increased dynamic power consumption in the proposed multistandard channelization accelerator is primarily contributed by the programmable FIR filter in the last stage. This stage decimates by a factor of two, and is implemented in polyphase form, with each subfilter mapped to a single MAC FIR filter. Temporal style FIR filter implementations experience significantly higher switching activity when compared to spatial style implementations due to loss of input correlation, resulting in increased dynamic power. A number of power optimizations strategies like coefficient reordering and coefficient segmentation have targeted the reduction of switching activity in MAC FIR filters [139]. Other works have focused on using increased parallelism and algorithmic strength reduction for reducing the power consumption of programmable time-shared filters [140, 141]. The use of these optimizations can further reduce the power consumption gap between the single mode accelerators and the proposed multistandard accelerator. The proposed design is scalable in terms of area when the number of standards are increased as evident from Table 4.3.

When compared to the filter coprocessor approach, the proposed accelerator will always be more power efficient, as it offloads a significant portion of the computational load from generic MAC units to hardwired filter stages, which can exploit constant coefficient optimization algorithms to perform the coefficient multiplications at a very low power consumption. The power consumption of the coprocessor method is highly dependent

4.4 Experimental Synthesis Results

on a number of architectural parameters: the degree of parallelism, the temporal correlation effects due to time multiplexing, the number of memory accesses, the ability to exploit linear phase property and polyphase decomposition, the control overheads, etc. A coarse architecture-independent estimate of a coprocessor power consumption can be obtained using the information about the number of MAC operations per second (MAC/s) in a particular mode and the energy per operation of a generic MAC unit. The computational load of the channelization accelerator (in terms of million MAC operations per second (MMAC/s) has been listed for different operational modes in the Table 4.4. Note that the tabulated computational load represents the decimation filtering load on either the I or Q branch of the DFE. The total computational load will therefore be twice of these values.

Table 4.4: Channelization Computational Load

Standard	Channelization Computational Load
GSM	439.118 MMAC/s
W-CDMA	1502.08 MMAC/s
IEEE 802.11a	3872 MMAC/s
WiMAX	5052.57 MMAC/s

A MAC circuit comprising of a 16x16 bit multiplier and a 32-bit accumulator was synthesized using the TSMC 0.18 μ m process. The energy of a single MAC operation was estimated using the Synopsys Design Compiler as 164.268 pJ. Using the computational load data from Table 4.4, the estimated power consumption of the coprocessor approach is shown in Table 5.1, along with the power consumption of the Velcro style accelerator and the proposed multistandard accelerator.

Table 5.1 reveals that the proposed multistandard accelerator significantly outperforms a filter coprocessor based accelerator implementation in terms of power consumption. The dynamic power consumption is slightly higher in the proposed accelerator, when compared to the Velcro approach due to the programmability requirements of the the NRCIC stage and the last stage filter. When compared to Velcro style multistandard accelerator, the proposed architecture has a greater potential for in-field upgrades for supporting a new standard, which is one of the important features envisioned in a

4.4 Experimental Synthesis Results

Table 4.5: Dynamic power consumption in different modes.

Standard	Velcro style accelerator	Proposed multistandard accelerator	Filter coprocessor
GSM	0.64 mW	0.9 mW	72.13mW
W-CDMA	2.07 mW	7.23 mW	246.74 mW
IEEE 802.11a	6.3 mW	13 mW	635.04 mW
WiMAX	7.2 mW	28.4 mW	830 mW

SDR. Note that the proposed factorization scheme implicitly assumes that the SRC factor, M_j , is greater than or equal to the factor $k = m_1 \cdot m_2 \dots m_n$. It is usually possible to identify a set of factors, m_1 to m_n such that k is smaller than M_j , for all the standards under consideration. For instance, it can be seen that for the four standards considered here, M_j , is greater than $k = 4$. The presence of the NRCIC stage and the transpose Farrow stage, which handle the standard dependent rational decimation load of k_j , allows any arbitrary SRC factor, greater than or equal to 4 to be factorized into the regular form required by Eq. 4.12. Hence, the reusability of the hardwired NRCIC stage, the transpose Farrow stage and the halfband stage in the proposed multistandard channelization accelerator for a new standard is limited only by the following two factors:

- (1) The SRC factor is greater than 4.
- (2) The hardwired filter stages satisfy the alias attenuation requirement of the new standard.

The fixed decimation load, k , after the fractional decimation stage is a design parameter that has to be chosen by the designer after taking into consideration, the $\Sigma\Delta$ ADC architecture, the expected SRC factors for different standards and the alias attenuation offered by the chosen fractional decimation stage.

Reusing the above multistandard accelerator for a new standard can be illustrated by the following example. Assume that the SDR handset has to be upgraded to support the Long Term Evolution (LTE) standard (20 MHz bandwidth) [142]. The 20 MHz LTE channel bandwidth is identical to the channel bandwidths of WiMAX and IEEE 802.11a standards, and can reuse the same multimode $\Sigma\Delta$ ADC [30] used for the above design. The symbol rate corresponding to the 20 MHz channel is 30.72 MSymbol/sec. The resultant SRC factor can be mapped onto the multistage decimation filter as

4.4 Experimental Synthesis Results

shown in Table 4.6. It can be seen that the NRCIC stage is completely bypassed in the LTE mode. The configuration information for bypassing the NRCIC stage, the configuration information for the Farrow structure and the coefficients for the last stage programmable FIR filter have to be supplied through an infield-upgrade. The hardwired halfband stage can be reused without any reconfiguration overhead.

Table 4.6: Factorization of SRC Factor for LTE

Standard	Sampling rate	Symbol rate	NRCIC	Farrow	Halfband	FIR
LTE	160 MHz	30.72 MSymbol/sec	-	1.3021	2	2

Benchmarking the proposed multistandard accelerator directly against other decimation filter structures is difficult due to difference in the assumptions about $\Sigma\Delta$ ADC sampling rates and output sample rates, presence or absence of a fractional SRC stage and pulse shaping stages within in the decimation chain, and differences in the coverage of standards. Instead, The computational complexity of the proposed multistandard accelerator, evaluated in Table 4.4 could be used to estimate the power consumption when the same computational load is performed using an alternate filter design style. A software implementation on a DSP, a coarse grained array implementation using MONTIUM tiles, and a filter coprocessor implementation using a high-speed energy-efficient 2-cycle MAC unit are used for benchmarking the proposed accelerator in the discussion below.

Texas Instruments’s TMS320C6472 processor is among the current state of art commercial multicore VLIW DSP processors targeted at high performance computing applications [143]. Its quoted energy efficiency is 0.15 mW/MMAC/s [144] (Note that ‘mW/MMAC/s’ has the same dimensions as energy and $0.15 \text{ mW/MMAC/s} = 150 \text{ pJ}$). The MONTIUM tile, whose architecture was discussed in Section 2.5, has an energy efficiency of 0.6 mW/MHz [145]. Once configured, the MONTIUM tile does not incur any In each cycle, the MONTIUM tile can perform 5 16-bit MAC operations in parallel, hence the energy efficiency for MAC operations is 0.12 mW/MMAC/s. The filter coprocessor used for benchmarking in Table 5.1 used a MAC structure built from the Synopsys library components. More efficient MAC structures have been reported in the academic literature. A high-speed energy efficient, 2-cycle MAC unit having an

4.4 Experimental Synthesis Results

energy efficiency of 13.8 pJ per MAC operation was reported in [146]. All the above three platforms are implemented on very different technologies and supply voltages. To provide a common basis for comparison, their energy efficiencies should be normalized to a common technology and supply voltage. The energy of a CMOS circuit is linearly dependent on the capacitance and quadratically dependent on the supply voltage. The capacitance in turn is linearly dependent on the technology node size. This observation was used to normalize the energy efficiencies of the above platforms to that of a 180 nm node with supply voltage of 1.8 V, as shown in Table 4.7. For instance, the energy of 90nm, 1V circuit when implemented in a 180nm, 1.8V technology will increase by a factor of $\left(\frac{180}{90}\right) \times \left(\frac{1.8}{1}\right)^2$.

Table 4.7: Energy of MAC operation of three alternative implementation styles

Platform	Technology, VDD	Reported Energy	Normalized Energy (180nm, 1.8V)
TMS320C6472	90nm, 1V	150 pJ	972 pJ
MONTIUM	130nm, 1.2V	120 pJ	373.85 pJ
2-cycle MAC	65nm, 1.1V	13.8 pJ	102.328 pJ

Using the energy efficiency values in Table 4.7 and the channelization accelerator computational load tabulated in Table 4.4, the power consumption for the different design styles can be evaluated as follows.

Table 4.8: Dynamic power consumption in different modes.

Standard	Proposed multistandard accelerator	TMS320C6472 DSP	MONTIUM processor	2-cycle MAC based filter coprocessor
GSM	0.9 mW	426.82 mW	164.16 mW	44.93 mW
W-CDMA	7.23 mW	1460.02 mW	561.55 mW	153.70 mW
IEEE 802.11a	13 mW	3763.58 mW	1447.54 mW	396.21 mW
WiMAX	28.4 mW	4911.09 mW	1888.90 mW	516.78 mW

Table 4.8 gives a fair idea of the power consumptions to be expected across a range of implementation platforms for the DFE: a software implementation on a DSP, a

4.5 Clocking and Runtime Parametrization Issues in the Multistandard Channelization Accelerator

coarse grained reconfigurable array, a filter coprocessor and the proposed multistandard accelerator, which is essentially a parameterizable ASIC. The DSP based DFE implementation suffers from the lowest energy efficiency, due to the instruction fetch and decode cycle overheads inherent in a software implementation. The MONTIUM coarse grained array behaves like an ASIC, after it has been configured for a particular mode. However the PEs in the array are still generic in nature and are not specialized for the DFE filtering tasks. The power consumption 2-cycle MAC based filter coprocessor gives an idea of the extent of power reduction achievable by low level arithmetic optimizations of the MAC unit. It can be seen that the resulting power consumption is still substantially larger than the proposed multistandard accelerator, which offloads a significant portion of the computational load to hardwired filter stages.

4.5 Clocking and Runtime Parametrization Issues in the Multistandard Channelization Accelerator

In this section, we discuss some of the implementation issues which stem from the need to provide multiple clock signals in a multistandard channelization accelerator and the need for evolving a practical parametrization strategy for configuring the above accelerator for a new mode.

4.5.1 Clocking

The practical problem of supplying multiple clock signals has not been studied in most of the existing multistandard multistage decimation filter designs [62, 78, 80, 81]. Unlike other hardware accelerators, clocking in a spatial style implementation of a multistage decimation filter is complicated by the fact that each stage operates at a different clock rate. A particular filter stage may also operate at different clock rates for different standards. The multiplicity of clocks can act as a major bottleneck for multistandard support in a resource constrained mobile handset. Traditionally the clock signals in a wireless mobile handset are generated from a reference frequency obtained from a temperature compensated crystal oscillator (TCXO) due to their stability, accuracy and good temperature characteristics [147]. However providing multiple TCXOs for generating multiple reference signals in a multistandard radio is unfeasible and not scalable when a large number of standards need to be supported. A more practical

4.5 Clocking and Runtime Parametrization Issues in the Multistandard Channelization Accelerator

solution is to use a fractional-N phase-locked loop (PLL) to generate the master clock for each standard from a single reference signal [148]. Unlike traditional integer-N PLLs, which have only an integer frequency divider in the PLL feedback loop, the fractional-N PLL frequency divider can assume rational values. This allows more clock signals to be generated from a single reference signal.

Assuming that the output of the fractional-N PLL (henceforth referred to as the master clock) has been chosen such that it is the least common multiple (LCM) of all the required clocks in the multistage decimation filter, the clocks for each stage can be obtained by a very simple direct digital synthesizer (DDS). The simplest DDS is the pulse output DDS which comprises of just an adder and a register as shown in Fig. 4.11 [149]. The output clock, F_{out} of the pulse output DDS can be given as follows.

$$F_{out} = \frac{F_{clk} \times F_{cw}}{2^N} \quad (4.27)$$

where F_{cw} is the frequency control word which controls the output frequency. In general,

Figure 4.11: Pulse output DDS.

the pulse output DDS has very high jitter and spur levels except for output frequencies which are exact integer submultiples of the input clock. However the expression in Eq. 4.27 cannot represent many integer submultiples. The above architecture can be made more flexible by incorporating a comparator in addition to the adder and register as shown in Fig. 4.12.

The modified structure is essentially an up-counter whose accumulated value is reset whenever it is equal to the frequency control word. The comparator output is used as

4.5 Clocking and Runtime Parametrization Issues in the Multistandard Channelization Accelerator

Figure 4.12: Clock enable counter.

the output clock pulse. The structure acts as a clock enable circuit which enables every F_{cw} th input clock pulse. Thus the output clock can be given as follows.

$$F_{out} = \frac{F_{clk}}{F_{cw}} \quad (4.28)$$

However the use of such a simple clock enable circuitry for generating the clock signals for the filter stages requires the master clock to be the LCM of all the required clocks in the multistage decimation filter. The clock rates of the stages prior to the fractional decimation stage are integer submultiples of the $\Sigma\Delta$ ADC sample rate, while the clock rates of the stages after the fractional decimation stage are multiples of the symbol rate. The $\Sigma\Delta$ ADC sample rate is only a function of the channel bandwidth and is independent of the symbol rate of the standard. This can lead to impractically high values for master clock. For instance, in WiMAX mode, the LCM of the clock rates of all the stages in the proposed accelerator is 41.76 GHz. A simple method to avoid such high master clocks would be to have two different fractional-N PLLs for generating the master clocks corresponding to the stages before and after the fractional decimation stage respectively. However this method incurs the additional area and

4.5 Clocking and Runtime Parametrization Issues in the Multistandard Channelization Accelerator

power penalty due to the extra PLL. A more efficient strategy would be to exploit the fact that if the reference frequencies of the fractional-N PLL is properly chosen, an additional master clock could be derived from it without needing another PLL. Most of the reported multistandard $\Sigma\Delta$ ADCs reported in literature [48] choose from a fixed set of sampling frequencies which are a function of the channel bandwidth and the degree of oversampling required in the particular $\Sigma\Delta$ ADC configuration. Since the range of channel bandwidths supported by the $\Sigma\Delta$ ADCs is known a priori, the reference frequency can be chosen as the LCM of the $\Sigma\Delta$ ADC sample rates. For instance, in the $\Sigma\Delta$ ADC considered in Section 4.4, the LCM of the supported sampling rates is 320 MHz. Hence if the reference frequency of the fractional-N PLL is fixed at 320 MHz, this signal can be used as the master clock for deriving the clock rates for the stages prior to the fractional decimation stage in all the standards. The output of the fractional-N PLL can be used as the master clock for deriving the clock rates for the stages after the fractional decimation stage. Thus one of the master clocks is fixed and hardwired for all standards, the second master clock is variable and is a function of the standard specific symbol rate. The proposed clocking arrangement is illustrated in Fig. 4.13.

Figure 4.13: Clock enable counters for the proposed multistandard channelization accelerator.

4.5.2 Runtime Parametrization and the Reconfiguration Bitstream

So far we have not yet addressed the actual mechanism for switching in a new configuration context for parameterizing the channelization accelerator for a mode. There are a number of functional units which needs to be supplied with some parameters at runtime.

- The steering logic which controls the number of NRCIC filter stages
- The fractional sample interval parameter of the transpose Farrow structure
- The frequency control words for the clock enable counters
- The filter coefficients for the programmable filters in the last stage

This translates to a considerable amount of reconfiguration data. These parameters and control signals have to be valid for as long as the accelerator is operating in a particular mode. The controller managing the reconfiguration process can be expected to have a limited memory bandwidth. Thus, when switching to a new mode, the configuration information needs to be loaded into the accelerator across multiple clock cycles. This poses two important design problems. Firstly, a mechanism has to be devised that allows the configuration information to be stored locally in the accelerator. Secondly, the configuration data has to be formatted as frames, each of which can be loaded by the reconfiguration controller in a single clock cycle.

Two kinds of configuration information need to be distinguished. The parameters controlling the NRCIC stage, the transpose Farrow structure and the clock enable counters need to be simultaneously available, throughout the period of operation. Only a limited number of filter coefficients need to be simultaneously available at any one point of time, the number being a function of the number of parallel MAC units the programmable last stage filter. The latter requirement can be handled by using of a reloadable random access memory (RAM) or a register file for storing the filter coefficients. The former requirement can be handled by storing the configuration data in a tapped delay line structure as shown in Fig. 4.14.

Configuration information can be loaded into the tapped delay line by setting the configuration write enable signal (Config-WE) high. When Config-WE is low, the tapped

4.5 Clocking and Runtime Parametrization Issues in the Multistandard Channelization Accelerator

Figure 4.14: Tapped delay line based configuration memory.

delay line holds the configuration information until the configuration reset signal (Config_RS) is enabled. Multiple taps supply the configuration data to the blocks which need the configuration information simultaneously. The channelization accelerator core showing all the input and output data ports as well as the configuration input ports is shown in Fig. 4.15. Reloading of the coefficient memory in the last stage programmable filter can be achieved using the two control signals, coefficient memory load (Coeff_LD) and coefficient memory write enable (Coeff_WE). The Coeff_LD signal going high triggers the coefficient reload process. The Coeff_WE signal now remains high for N_{size} clock cycles, where N_{size} is the size of the coefficient memory. The wordlengths of the last stage filter coefficients, the frequency control word of the clock enable generator and the Farrow structure control word are set to 16 bits. The four stage NRCIC filter requires 4-bits to control the number of stages. The configuration frames for the channelization accelerator are arranged as shown in Fig 4.16. The total bitstream is formatted as 20-bit wide frames, each of which are loaded into the accelerator in one clock cycle. The fixed master clock is used for clocking the tapped delay line. Each frame has 4 control bits which determine if the 16-bit data payload should be interpreted as a configuration data which needs to be loaded into the configuration tapped delay line or a filter coefficient which needs to be loaded into the coefficient memory. A total of 10 frames are required to configure the tapped delay line after it is reset. This provides the configuration parameters for the clock enable counters, the NRCIC stage and the transpose Farrow structure. The number of frames required for configuring the programmable filter is a function of the size of the coefficient memory.

4.5 Clocking and Runtime Parametrization Issues in the Multistandard Channelization Accelerator

Figure 4.15: The channelization accelerator architecture.

Figure 4.16: The reconfiguration bitstream.

The above mechanism for runtime parameterizations was verified using an experimental simulation. A model based HDL design of the tapped delay line based configuration memory, shown in Fig. 4.14 as created using the Xilinx System Generator. Xilinx

System Generator is a DSP design tool that offers an alternative to the traditional behavioural HDL based FPGA design flow. The design entry in System Generator takes advantage of the model-based MATLAB Simulink design environment. System Generator itself is integrated into Simulink as a blockset (toolbox) comprising of fundamental DSP operators, memories as well as more complex Xilinx IP cores, which can be directly instantiated on a FPGA during the architectural synthesis. Each frame in the reconfiguration bitstream in Fig. 4.16 comprises of the 1-bit inputs to the Config_WE, Config_RS, COEFF_LD, and COEFF_WE input ports, and the 16-bit input to the configuration input port, as shown in Fig. 4.15. Test vectors for each of these ports were generated separately. The Xilinx ISIM environment was used to stimulate the model under test, and observe the waveforms. Fig. 4.17 indicates the sequence of control signal that are used for parameterizing the accelerator. The Config_RS signal initiates the runtime parametrization of the accelerator. It resets all the configuration registers and initializes them with their default values. The parameterization occurs in two distinct steps which can be performed in any order : reloading of the programmable filter coefficient memory, and the reloading of the tapped delay line based configuration memory. The tap values have been shown in Fig. 4.18 at different stages of the parameterizations. It can be seen that after Coeff_WE signal is enabled for 10 consecutive clock cycles, the tapped delay line is completely configured with the clock enable control words, the NRCIC stage bypass control word and the Farrow's structure control word.

Figure 4.17: The control flow for parameterizing the accelerator

4.6 Summary

Emerging communication paradigms require radio terminals to provide seamless mobility across multiple access networks. HW accelerators which implement the com-

Figure 4.18: Loading of the tapped delay line based configuration memory

putationally intensive baseband kernels are typically optimized for a single standard, and hence, may limit the flexibility of a multistandard SDR. A traditional Velcro style accelerator implementation of the computationally intensive channelization function is not scalable for supporting an arbitrary number of standards. Reducing the silicon area penalty for providing multistandard support requires the opportunities for hardware reuse to be identified across multiple standards. In a single mode channelization accelerator, the filter stages which are a part of the multistage decimation filter which performs the channelization function, can be efficiently implemented at a low area/power cost with the help of constant-coefficient filter optimizations. However the ability to reuse these hardwired filter stage for a different standard is limited by the dependencies on the standard specific band-edge specifications, the attenuation requirements and the factorization of the SRC loads.

In this chapter, we have proposed a new factorization strategy to split any arbitrary SRC loads onto a fixed set of decimation filter stages. The proposed method has the advantage that the stages prior to the last stage have very low reconfiguration overheads, and can be readily reused across multiple standards. Only the last stage FIR filter needs to be reprogrammable for supporting variable coefficient sets and filter lengths. The reusable stages prior to the last stage also benefit from the area reductions obtained by constant-coefficient filter optimizations. The proposed multistandard decimation filter thereby has a silicon area which is close to that of a single-mode decimation filter, while still being scalable enough to support an arbitrary number of standards.

4.6 Summary

Since the proposed structure offloads a significant portion of the computational load to hardwired filter stages, it has a significantly lower dynamic power consumption than a filter coprocessor implementation where all the coefficient multiplications have to be performed on generic MAC units. We have also discussed how clock signals can be provided to the accelerator and how it can be reconfigured at runtime using a specially designed configuration architecture and reconfiguration bitstream.

5

Design of Power Efficient Programmable Time-shared Filters

The channelization accelerator proposed in Chapter 4 has a significantly lower dynamic power consumption when compared to a filter coprocessor, since it offloads a significant amount of the computational load from generic MAC units to hardwired filter stages. These stages perform the coefficient multiplications using a network of adders and hardwired shifts. Since the hardwired stages are reused across multiple standards, the proposed structure has a lower area overhead than a Velcro style accelerator. The lower area translates to lower number of leaking transistors and reduced leakage power. However the last stage filter in the proposed multistandard channelization accelerator still needs to be reprogrammable for accommodating varying band-edges and pulse shaping factors, which necessitates the use of generic MAC units. The power consumption of the proposed multistandard accelerator and the Velcro accelerator is recalled in Table 5.1.

The main factor which contributes to the higher power consumption of the proposed multistandard accelerator when compared to the Velcro accelerator is the presence of generic multipliers in the last stage programmable filter. It is currently implemented in polyphase form, with each subfilter implemented using a single MAC unit. Generic multipliers have a significantly higher area and power penalty when compared to con-

Table 5.1: Dynamic power consumption in different modes.

Standard	Velcro style accelerator	Proposed multistandard accelerator	% Increase in power consumption due to reprogrammability
GSM	0.64 mW	0.9 mW	40.62%
W-CDMA	2.07 mW	7.23 mW	249.29%
IEEE 802.11a	6.3 mW	13 mW	106.35%
WiMAX	7.2 mW	28.4 mW	294.44%

stant multipliers, which can be implemented using adders and hardwired shifts. The goal of this chapter is to further reduce the power consumption gap between the proposed multistandard accelerator and a Velcro style accelerator, by devising strategies for reducing the area/power penalties of the programmable filter used in the last stage. Programmable filters have been generally implemented in a temporal fashion. The regularity of FIR filter data flow graph can be exploited for multiplexing the coefficient multiply operations onto a fixed set of MAC units with minimal control overheads. The regularity also allows arbitrary filter lengths to be mapped onto a fixed set of MAC units [121]. However, when compared to spatial style implementations, temporal implementation suffer from increased switching activity, due to loss of input correlation. Most of the prior works on reducing the power consumption of time-shared filters, have focused on reducing the increased switching activity [124, 150, 151, 152]. In the nanoscale CMOS technologies, the leakage component has started dominating the overall power consumption. Hence the filter power optimization strategies should also take the contribution due to leakage power into account. In this chapter we investigate parallelism as a power reduction strategy for programmable time-shared filters in nanoscale CMOS. Increasing parallelism may seem like a counterintuitive strategy since leakage power is strongly correlated to the total gate width, which in turn is proportional to the area [95]. However, in a fixed throughput system, increased parallelism lowers the operating frequency and increases the timing slack of the MAC units. In nanoscale CMOS technologies, this slack can be utilized for implementing the circuits using transistors having thicker gate oxides and higher threshold voltages, which are slower but have the advantage of having lower leakage currents. Since we are exploiting the power reductions achievable due to increased timing slacks to offset the leakage

5.1 Parallelism and Power Consumption in Nanoscale CMOS Circuits

power and area penalty of parallelism, one important architectural level design consideration is the degree to which a particular structure can trade area for timing slack. In this chapter, we propose a class of programmable time-shared FIR filters based on fast filter algorithms [33], that can trade area for increased timing slack, more efficiently than traditional direct form based time-shared filters, up to certain filter lengths. The range of filter lengths over which the proposed structures are more power efficient increases with increasing levels of parallelism. The increased timing slack provides more room for supply and threshold optimizations, which can be used to reduce all the major nanoscale CMOS power consumption components.

5.1 Parallelism and Power Consumption in Nanoscale CMOS Circuits

5.1.1 Trading area for timing slack

One of the important architectural-level design parameters in the design of a hardware accelerator is the degree of parallelism. Parallelism has been traditionally used as an architectural transformation for trading increased area for reduced dynamic power [100]. In higher device geometries, CMOS power consumption is largely dominated by dynamic power consumption, which is caused by the charging and discharging of capacitive nodes. We recall the expression for dynamic power below [100]:

$$P_{dyn} = \alpha_f C_T V_{DD}^2 f_{clk} \quad (5.1)$$

where C_T stands for the total capacitive load, V_{DD} stands for the supply voltage, α_f is the fraction of capacitive nodes that are switching and f_{clk} is the clock frequency. Eq. 5.1 reveals a quadratic dependence on the supply voltage and a linear dependence on the clock frequency. In a fixed throughput system, increasing the level of parallelism has the effect of reducing the frequency of operation and hence increasing the timing slack in the critical path. The delay of a CMOS circuit depends on the time taken to charge and discharge the capacitive nodes, and can be expressed using the alpha-power law MOSFET model [154], as follows:

$$T_{pd} = \frac{K_D V_{DD}}{(V_{DD} - V_{th})^\alpha}, \quad (5.2)$$

5.1 Parallelism and Power Consumption in Nanoscale CMOS Circuits

where K_D is a fitting parameter, V_{th} is the threshold voltage, and α is the velocity saturation term, which is equal to 1.5 in short channel transistors. Eq. 5.2 suggests that any available timing slack in the circuit can be exploited for reducing the supply voltage or for increasing the threshold voltage. In higher device geometries, the quadratic dependence of dynamic power on supply voltage makes supply voltage scaling a useful strategy for exploiting the parallelism-induced timing slacks. The physical capacitance term, C_T , in Eq. 5.1 increases due to increased parallelism. However this increase is easily compensated by the quadratic reduction that stems from the lower supply voltage. Keeping the circuit delay constant, the supply voltage can be further lowered by lowering the threshold voltage. Lowering the threshold voltage results in an increase in the gate overdrive. This dual scaling of both supply and threshold voltages yields greater reductions in the dynamic power, than achievable by supply voltage scaling alone.

The traditional architecture-driven supply/threshold voltage scaling model, shown in Fig 5.1, faces some serious bottlenecks in nanoscale technologies. Aggressive technology scaling over the years has led to the leakage power replacing the dynamic power as the dominant source of power consumption even in the active mode and not just the standby modes. A reduction in the supply voltage causes a quadratic reduction in the dynamic power and an exponential reduction in gate leakage, however lowering the threshold voltage causes an exponential increase in the subthreshold leakage current. Both the gate leakage and the subthreshold leakage are also linearly dependent on the gate width. Hence the total leakage power consumption is strongly correlated with the total gate width and area [95]. The total physical capacitance term, C_T , in Eq. 5.1 is also strongly correlated with the total area. Hence, a parallelism based power reduction strategy for nanoscale CMOS technology nodes should ensure that the potential gains from increased timing slacks are not offset by the area penalty.

Parallelism reduces the operating frequency, f , of the datapath operators. Assuming that the supply and threshold voltages are scaled such that the circuit delay is equal to the cycle period, the operating frequency can be expressed in terms of the supply and threshold voltages as follows:

$$\frac{1}{f} = \frac{K_D V_{DD}}{(V_{DD} - V_{th})^\alpha} \quad (5.3)$$

5.1 Parallelism and Power Consumption in Nanoscale CMOS Circuits

Figure 5.1: Traditional supply/threshold voltage scaling.

Eq. 5.3 can be reformulated as follows:

$$V_{th} = V_{DD} - (L_D K_D V_{DD} f)^{\frac{1}{\alpha}} \quad (5.4)$$

Eq. 5.4 can be used to plot the locus of permissible (V_{th}, V_{DD}) pairs for a fixed frequency constraint. To demonstrate how the above locus shifts with varying frequency constraints, we synthesized a 16 bit adder in a TSMC $0.18\mu\text{m}$ process, and used it to extrapolate the circuit specific parameters, L_D and K_D . Fig. 5.2 shows the locus of (V_{th}, V_{DD}) duplets having a fixed frequency, for the 16 bit adder. It shows that relaxed frequency constraints can be used to lower the supply voltages and increase the threshold voltages. The higher V_{th} , lower f , and lower V_{DD} values result in a cubic reduction of dynamic power, and an exponential reduction of subthreshold leakage power and gate leakage power. These reductions can compensate the near-linear increase in total power due to increased area. Thus reducing the nanoscale CMOS power consumption components requires the threshold voltage to be scaled up, and the supply voltage to be scaled down, as shown in Fig. 5.3

5.1.2 Interplay between architectural and device level power optimizations

Increasing the level of parallelism shifts the permissible (V_{th}, V_{DD}) values to the high V_{th} , lower V_{DD} region. A lower V_{DD} reduces the voltage drop across the gate oxide, resulting in an exponential reduction in gate leakage currents. The higher permissible V_{th} values permit the use of thicker gate oxides. The gate oxide capacitance per unit area, C_{ox} , can be expressed in terms of the gate oxide thickness, T_{ox} , as follows:

$$C_{ox} = \varepsilon_{ox} / T_{ox}, \quad (5.5)$$

5.1 Parallelism and Power Consumption in Nanoscale CMOS Circuits

Figure 5.2: Locus of (V_{th}, V_{DD}) points having a fixed frequency constraint.

Figure 5.3: Voltage scaling for nanoscale CMOS technologies.

Using Eq. 5.5, the relationship between threshold voltage and the gate oxide thickness can be shown as follows.

$$V_{th} = V' + \frac{T_{ox} \sqrt{2\epsilon_{si} q N_A (2\psi_B + V_{SB})}}{\epsilon_{ox}}, \quad (5.6)$$

$$V' = V_{FB} + 2\psi_B - \eta V_{DS}. \quad (5.7)$$

where V_{FB} is the flatband voltage, V_{SB} is the body bias, V_{DS} is the drain-to-source voltage, N_A is the doping density, ϵ_{si} is the permittivity of silicon, q is the electronic charge and ψ_B is the difference between the Fermi potential and substrate intrinsic

5.1 Parallelism and Power Consumption in Nanoscale CMOS Circuits

potential. It can be seen from Eq. 5.6 that the threshold voltage, V_{th} is linearly dependent on T_{ox} . When T_{ox} is increased, the threshold voltage increases, resulting in an exponential reduction in subthreshold leakage. Increasing the gate oxide thickness reduces the gate capacitance, which in turn reduces the dynamic power consumption. Increased T_{ox} also causes an exponential reduction in the gate leakage power. Thus increasing the gate oxide can help reduce all the nanoscale CMOS power consumption components.

Forward body bias can be used for further increasing the thickness of the gate oxide. A forward body bias for n-channel transistors occurs when $V_{SB} < 0$. It can be used as an alternate leakage reduction strategy in which the performance of nominally high V_{th} devices, with a low standby leakage power can be improved when a forward body bias is applied in the active mode. One important advantage of this technique is that it can scale well into lower geometries because of reduced short channel effects under the forward body biased condition [155]. Traditional standby leakage techniques like power gating and reverse body bias do not perform well in nanoscale CMOS [102, 157]. The threshold voltage is also more sensitive to a forward bias voltage. The application of a forward body bias allows the gate oxide thickness to be increased further, while maintaining a given threshold voltage. However, the magnitude of the applicable forward bias is limited to around 600mV due to the need for maintaining a reverse bias at the drain-body and source-body p-n junctions of the MOSFET.

The underlying idea of using thicker gate oxides to reduce the leakage currents has been traditionally used in a multiple oxide CMOS process [102]. In that process, the non-critical paths are implemented using low performance, high T_{ox} transistors whereas the critical paths are implemented by using high speed transistors with a lower T_{ox} . When the number of critical paths is much lower than that of the non-critical paths, this technique significantly reduces the leakage power consumption. However, the drawback of the technique is that it requires a complicated technological process for fabrication. The combined effect of forward body bias and increased timing slacks due to increased parallelism now allows the whole circuit to be implemented using a thicker gate oxide, than would have been possible with only either of the two methods. During the active mode, the reduced supply voltage, reduced frequency of operation and thicker gate oxides reduce the dynamic as well as the leakage power consumption components. In the standby mode, the removal of forward body bias results in a significant reduction of

5.2 Area Efficiency of Time-shared FIR Filters

the leakage currents. Thus, using the architectural driven voltage scaling together with forward body bias reduces the leakage and dynamic power consumption components without having to resort to an advanced CMOS process.

5.2 Area Efficiency of Time-shared FIR Filters

All the three major power components in nanoscale CMOS, show a near-linear dependence on the area. The physical design level power minimization strategy, discussed in the last section, depends strongly on the amount of timing slack that can be obtained by increasing the area. Aggressive scaling brings the supply and threshold voltages closer to each other. In Fig. 5.4, the normalized delay, $T_N = V_{DD}/(V_{DD} - V_{th})^{1.5}$, is plotted against the V_{th} , where $V_{DD} = 1V$ and $V_{DD} = 0.9V$. It can be seen that when the V_{th} approaches V_{DD} , the delay curve climbs steeply. This behavior implies that even a small change in V_{th} , causes a large increase in delay. The use of parallelism in this region incurs a large area penalty. Hence, one important architectural-level design parameter, for using parallelism in nanoscale technologies is the efficiency with which a parallel architecture trades area for timing slack. Finding a suitable architecture, that offers a large increase in timing slack for a small increase in area is still an open problem. To measure this trade-off, we define the metric, ‘area-slack efficiency’ as the amount of timing slack increment obtained per unit increase in area. In this chapter, we discuss two different classes of time-shared filters, traditional time-shared filters derived from direct-form FIR filter structures and a novel class of time-shared filters derived from fast filter algorithm (FFA) structures, and compare their area efficiency. Finally the area/power performance of these structures against varying levels of parallelism is analyzed.

5.2.1 Direct Form Structure based Time-shared Filters

The data flow graph of a direct form structure is highly regular. Hence a time-shared FIR filter structure can be derived by multiplexing the coefficient multiplications onto a limited set of MAC units, as shown in Fig. 5.5. Consider a direct form filter of length N , which is folded onto M MAC units (M is typically much smaller than N , and is assumed to be an exact integer submultiple of N for simplicity). A coefficient buffer/data buffer of size $2N/M$ are required for each MAC unit, and can be implemented using a dual port

5.2 Area Efficiency of Time-shared FIR Filters

Figure 5.4: Normalized delay as a function of the threshold voltage.

RAM. The regularity of the direct form DFG allows the control logic to be implemented using simple counters. Assuming that the throughput requirement of the filter is f_{clk} , the operating frequency of each MAC can be given by Nf_{clk}/M . We use the notation $DF|M$ to indicate a direct form filter with M MAC units. The time-shared filter performs N MAC operations per output sample.

5.2.2 FFA based Time-shared Filters

5.2.2.1 Fast Filter Algorithms

FFA structures are a special class of parallel FIR filter structures that have the property of algorithmic strength reduction, i.e., they have a lower number of expensive MAC operations at the cost of increased add operations [33]. A K -parallel FIR structure, which corresponds to a N tap FIR filter, is a block processing structure that processes K inputs in parallel and produces K outputs. This structure has K^2 subfilters, each of length N/K (where N is assumed to be a multiple of K for simplicity). When multiple inputs are processed in parallel, significant redundancies exist across the subfilters. This

5.2 Area Efficiency of Time-shared FIR Filters

Figure 5.5: Direct form structure based time-shared filter.

phenomenon is used by the FFA structures to reduce the number of subfilters. A K -by- K FFA (henceforth, referred to as a $K \times K$ FFA) has S_K subfilters, where S_K is typically much smaller than K^2 , and each subfilter is of length N/K . This reduction in the total number of MAC operations comes at the expense of A_K preprocessing/postprocessing adders. The input sample rate is f_{clk} , the input sample rate of each of the subfilters in the $K \times K$ structure is given by f_{clk}/K . The $K \times K$ FFA structure performs a total of NS_K/K MAC operations and A_K additions for generating K parallel outputs. The number of multiplications per output sample, N_m , and additions per output sample, N_a , can be given by Eq. 5.8 and Eq. 5.9 respectively.

$$N_m = \left(\frac{S_K}{K^2} \right) N \quad (5.8)$$

$$N_a = A_K/K \quad (5.9)$$

The algorithmic strength reduction property of FFAs stems from the fact that the term S_K/K^2 is less than one, and hence FFA structures have a lower number of MAC operations than a direct form FIR structure, at the expense of an additional A_K/K add operations. The simplest case of a 2×2 FFA is shown in Fig. 5.6.

5.2 Area Efficiency of Time-shared FIR Filters

Figure 5.6: 2x2 FFA.

Higher order FFAs can be obtained by cascading smaller FFAs [33]. We have limited our current analysis to 2-stage FFAs for simplicity. Assume that a $K \times K$ FFA has to be constructed where K is a composite number that can be factorized as $K = K_1 \times K_2$. Firstly a $K_1 \times K_1$ FFA is applied to the original N -tap FIR filter resulting in S_{K_1} subfilters and A_{K_1} preprocessing/postprocessing adders. Now a block of K_2 outputs of each subfilter is considered to transform it using a $K_2 \times K_2$ FFA. The resulting structure has $S_{K_1} \times S_{K_2}$ subfilters of length N/K each. The total number of preprocessing/postprocessing adders given by:

$$A_K = K_2 A_{K_1} + S_{K_1} A_{K_2} \quad (5.10)$$

A 4x4 FFA constructed by cascading two 2x2 FFAs is shown in Fig 5.7. The 2x2 FFA structure is repeated three times within the 4x4 FFA structure. In general, the $K_2 \times K_2$ FFA structure is repeated S_{K_2} times within the two stage $K \times K$ FFA structure. The S_K and A_K parameters for different values of K are shown in Table 5.2

Table 5.2: Number of subfilters and preprocessing/postprocessing adders for different FFA structures

K	S_K	A_K
2	3	4
3	6	10
4	9	20
5	12	40
6	18	42
8	27	76

5.2 Area Efficiency of Time-shared FIR Filters

Figure 5.7: 4x4 FFA.

5.2.2.2 Proposed Class of FFA based Time-shared Structures

In the 2x2 FFA structure, each subfilter is of length $N/2$, while each of the subfilters in the 4x4 FFA are of length $N/4$. The subfilters are either a polyphase component of the original filter, or an additive combination of different polyphase components. The general FFA structure consists of regular FIR subfilters and a highly irregular preprocessing/postprocessing addition network. The data flow graph of the resultant structure no longer has the regularity of traditional direct form structures. Hence time-multiplexing the arithmetic operations in the FFA structure onto a fixed set of MAC units and adders incurs a much higher control overhead. Temporal style dedicated ASIC implementations of FIR structures which exploit the algorithmic strength reduction for supply voltage scaling have been studied in [158], where the multiply and add operations are performed sequentially by a single multiplier and adder. The authors do not discuss the control overheads for the above structure. A more general class of time-shared filters can be obtained by time-multiplexing the regular FIR subfilters of the FFA structure onto a fixed number of MAC units, while the irregular preprocessing/postprocessing addition network DFG is directly mapped onto the hardware. Thus the subfilters are

5.2 Area Efficiency of Time-shared FIR Filters

implemented in a temporal style, while the preprocessing/postprocessing addition network is implemented in a spatial style. The regularity of the FIR subfilter DFG allows results in low control overheads. The resulting structure benefits from both the algorithmic strength reduction of FFAs and the flexibility to change the filter length and coefficient set offered by a temporal style implementation of the subfilters. Reprogramming the proposed FFA structure for a different coefficient set requires the coefficient memories of the FFA subfilters to be reloaded, while the preprocessing/postprocessing addition network remains unchanged. We use the notation ‘ $K \times K | L$ ’ FFA to indicate a time-shared $K \times K$ FFA structure in which each of the S_K subfilter is mapped onto L MAC units. A $2 \times 2 | 2$ FFA structure is shown in Fig. 5.8. With a throughput requirement of f_{clk} , the input sample rate for each subfilter is f_{clk}/K . Each of the subfilters is of length, N/K . If each subfilter is folded onto L MAC units, the operating rate of each MAC unit is Nf_{clk}/K^2L . The coefficient/data buffer size per MAC unit in this structure is given $2N/KL$. The preprocessing/postprocessing addition network operates at a rate of f_{clk}/K . The lower bound on the timing slack in the system and the bottleneck for V_{DD}/V_{th} scaling is still determined by the time multiplexed MAC units. The total number of MAC units is given by $M = S_K L$.

Figure 5.8: $2 \times 2 | 2$ FFA.

5.2 Area Efficiency of Time-shared FIR Filters

5.2.3 Comparison of Area Efficiencies

Let the critical path delay of a MAC unit for the nominal supply and threshold voltages be T_D . The timing slack in each MAC unit of a M MAC unit based FIR filter of length N can be given by $(M/Nf_{clk} - T_D)$. Increasing the level of parallelism by P MAC units, increases the timing slack to $((M + P)/Nf_{clk} - T_D)$. Increased parallelism increases the datapath area, while the coefficient/data buffer memory overhead remains unchanged. Assuming that the area of each MAC unit is A_m , the area penalty incurred due to the addition of P MAC units is PA_m . Hence, the area-slack efficiency of the folded direct form filter, E_{DF} can be expressed as follows:

$$E_{DF} = \frac{P}{Nf_{clk}} \times \frac{1}{PA_m} = \frac{1}{A_m Nf_{clk}} \quad (5.11)$$

The timing slack in each of the S_K MAC units of the $K \times K | L$ FFA structure can be given by $(K^2 L / Nf_{clk} - T_D)$. The $K \times K | L$ time-shared filter has low control overheads, as the same number of MAC operations are mapped to each of the $S_K L$ MAC units. This property allows same control logic to be shared by all the MAC units. Preserving the regularity of the structure while increasing the parallelism requires the number of MAC units in all the subfilters to be equal. Consider the case of a $K \times K | (L + P)$ time-shared filter, in which each of the S_K subfilters is mapped to $(L + P)$ MAC units. The additional MAC units incur a datapath area penalty of $S_K PA_m$, and increase the timing slack to $(K^2 (L + P) / Nf_{clk} - T_D)$. The area-slack efficiency, $E_{K \times K}$, of the FFA-based time-shared structures can be given as follows.

$$E_{K \times K} = \frac{PK^2}{Nf_{clk}} \times \frac{1}{S_K PA_m} = \frac{K^2}{S_K} \times \frac{1}{A_m Nf_{clk}} = \frac{K^2}{S_K} \times E_{DF} \quad (5.12)$$

The area-slack efficiencies can be compared by analyzing the K^2/S_K parameter in Table 5.3. For instance, it can be seen that the area efficiency of the 6x6 FFA, is twice that of the time-shared direct form filter. This implies that, $2P$ MAC units need to be added to a time-shared direct form filter, to obtain the same amount of timing slack increment induced by adding P MAC units to the 6x6 time-shared FFA structure. Hence, the area-slack efficiency metric can be used as an useful guideline, when parallelism is used as a power reduction strategy in time-shared filters.

5.2 Area Efficiency of Time-shared FIR Filters

Table 5.3: FFA Parameters.

K	S_K	A_K	K^2/S_K	S_K/K	K^2
2	3	4	1.33	1.5	4
3	6	10	1.5	2	9
4	9	20	1.78	2.25	16
5	12	40	2.08	2.4	25
6	18	42	2	3	36
8	27	76	2.37	3.375	64

5.2.4 Practical Limits for the Use of FFA Structures

We have seen that increasing the parallelism of the temporal style FIR subfilters in FFA structures yields greater increments in timing slacks than when the parallelism of time-shared direct form FIR filters is increased. However the following questions need to be answered before using FFA structures in a practical design.

- Does better area-slack efficiency always translate to lower power consumption ?
- Under what conditions are FFA structures more power efficient ?

Answering these questions requires an effective mechanism to compare the power consumption of the two paradigms for varying levels of parallelism. The area-slack efficiency metric gives an idea about the amount of timing slack increment induced by adding a given number of MAC units to a time-shared filter structure. In other words, it measures the sensitivity of the timing slack in different time-shared structures to increasing levels of parallelism. However this metric does not take into account, the ‘fixed’ area costs of coefficient/data buffering and pre-processing/post-processing adders which remain unchanged with varying levels of parallelism in the time-shared FIR subfilters of the proposed FFA structure. These costs need to be considered while evaluating the total area. Both the dynamic and leakage power consumption components in nanoscale CMOS are strongly correlated to the total area. It is shown in this section that, the coefficient/data buffering overhead of FFA based time-shared filters increases at a faster rate than that of folded direct form filters when the filter lengths are increased. As a result of their higher area-slack efficiency, FFA based time-shared filters require a lower number of MAC units than folded direct form filters, for a comparable amount of

5.2 Area Efficiency of Time-shared FIR Filters

timing slack. However, in high order filters, the higher memory overhead of FFA based time-shared filters might negate the advantage of their lower datapath area. Hence, it is important to identify the filter lengths up to which FFA based time-shared filters are more area efficient than folded direct form filters.

Consider the case of alternate time-shared filter implementations (both FFA based and direct form based) having an identical amount of timing slack (or equivalently, identical cycle periods) in the MAC units. The identical timing constraints allow identical supply and threshold voltages to be used for the different implementations. Note that the spatial style preprocessing/postprocessing adder network in a $K \times K | L$ FFA structure operates at the frequency f_{clk}/K while the MAC units and the memory units in the temporal style FIR subfilters operate at the frequency Nf_{clk}/KM . Hence the permissible supply and threshold voltages are determined by the timing slacks available in the MAC units. The total switching activity and dynamic power consumption of the time-shared FFA structure is also dominated by the temporal style subfilters. Under these conditions the total power consumption of the time-shared filters, including both the dynamic and leakage power consumption components can be expressed as a function of the designer-controlled parameters as follows.

$$P_{total} = F(A, f, V_{DD}, V_{th}) \quad (5.13)$$

The parameter A represents the area of the time-shared FFA structure. Both the total physical capacitance and the total gate width are strongly correlated to the total area. The parameter f represents the frequency of the MAC units in the subfilters. When the f , V_{DD} and V_{th} parameters are identical for alternate time-shared implementations, their power consumptions show a near-linear dependence on the total area. Hence the area of time-shared implementations having an identical amount of timing slack in the MAC units gives a measure of their power efficiency.

Assume that the area costs of the fundamental building blocks of a programmable FIR filter, namely a coefficient/data register, a MAC unit and an adder are given by A_r , A_m , and A_d respectively. The $K \times K | L$ FFA structure has $S_K L$ MAC units, with N/KL coefficient multiplications mapped to each MAC unit. The total datapath and coefficient/data storage area of this structure can be given as follows.

$$A_{K \times K | L}(N) = S_K L A_m + A_K A_d + 2 \left(\frac{S_K}{K} \right) N A_r \quad (5.14)$$

5.2 Area Efficiency of Time-shared FIR Filters

The operating frequency of the MAC units in the $K \times K | L$ FFA structure is $N f_{clk} / K^2 L$. To achieve the same operating frequency, a folded direct form filter should have $K^2 L$ MAC units with $N / K^2 L$ coefficient multiplications mapped to each MAC unit. The total datapath and coefficient/data storage area of the folded direct form filter with $K^2 L$ MAC units can be given as follows:

$$A_{K^2 L}(N) = K^2 L A_m + 2N A_r \quad (5.15)$$

The S_K , A_K , S_K/K , and K^2 values in Table 5.3 give an idea of about difference in the datapath and memory areas of the above two time-shared filter architectures. It can be seen from Eq. 5.14 and Eq. 5.15 that, the $K \times K | L$ FFA structure has a lower number of MAC units, but S_K/K times the coefficient/data memory area as that of the $K^2 L$ MAC based time-shared direct form filter. For short length filters, the filter area is dominated by the datapath area. With increasing filter length, the coefficient storage overhead starts dominating the overall area. For memory dominated time-shared filter structures, traditional direct form based structures are more power efficient than FFA based structures.

To estimate the values of A_m , A_d and A_r , a 16 bit register, a 32 bit adder circuit, and a MAC circuit comprising of a 16x16 bit multiplier and a 32 bit adder-accumulator were synthesized using a TSMC 0.18 μ m process. The Synopsys Design Compiler was used to estimate the cell area A_m , A_d and A_r respectively. Table 5.4 shows the area in terms of the gate count. These results were obtained by normalizing the area values by the cell area of a two input NAND gate of the same library.

Table 5.4: MAC, Adder and Register Area

Circuit	Gate Count
MAC(A_m)	2098
Adder(A_d)	223
Register(A_r)	85

Consider three alternate time-shared filter implementations- a time-shared direct form filter with 16 MAC units (denoted by the notation $DF|16$), a $2 \times 2 | 4$ FFA structure, and a $4 \times 4 | 1$ FFA structure. The MAC units in all the three implementations have

5.2 Area Efficiency of Time-shared FIR Filters

an operating frequency of $Nf_{clk}/16$. The area costs of the three structures are shown below.

$$A_{16|DF}(N) = 16A_m + 2NA_r \quad (5.16)$$

$$A_{2 \times 2|4}(N) = 12A_m + 3NA_r + 4A_d \quad (5.17)$$

$$A_{4 \times 4|1}(N) = 9A_m + 4.5NA_r + 20A_d \quad (5.18)$$

The above three functions have been plotted against varying filter lengths in Fig. 5.9. The A_m , A_r , and A_d values are taken from Table 5.4. It can be seen that higher order FFAs have the lowest area for low order filters. With increasing filter length, lower order FFAs are more efficient than higher order FFAs. When filter lengths are further increased, FFA structures yield diminishing returns due to increased coefficient/data memory overhead, and are less area efficient than traditional folded direct form structures.

Figure 5.9: Area of time-shared filters whose MAC units operate at $Nf_{clk}/16$.

With an increase in the level of parallelism in the time-shared filter, the relative contribution of the datapath operators is increased when compared to the coefficient/data memory area. Hence the filter length at which the area cost of the FFA structures

5.2 Area Efficiency of Time-shared FIR Filters

exceeds that of the folded direct form area, can be expected to increase. This is demonstrated in Fig. 5.10, which compares the area cost of a $DF|32$ structure, a $2 \times 2|8$ FFA structure, and a $4 \times 4|2$ FFA structure. The MAC units in all the three implementations have an operating frequency of $Nf_{clk}/32$. The expressions for the area costs of the three structures are shown below, and have been plotted in Fig. 5.10.

$$A_{DF|32}(N) = 32A_m + 2NA_r \quad (5.19)$$

$$A_{2 \times 2|8}(N) = 24A_m + 3NA_r + 4A_d \quad (5.20)$$

$$A_{4 \times 4|1}(N) = 18A_m + 4.5NA_r + 20A_d \quad (5.21)$$

Figure 5.10: Area of time-shared filters whose MAC units operate at $Nf_{clk}/32$.

It can be seen from Fig. 5.10, that filter lengths above which the $2 \times 2|8$ FFA has a lower area than the $4 \times 4|2$ FFA, and the direct form structure has a lower area than the $2 \times 2|8$ FFA is much longer than their corresponding counterparts in Fig. 5.9. The filter length, $N_{K \times K|L}$, above which the $DF|K^2L$ structure has a lower area cost than the $K \times K|L$ FFA structure can be derived from Eq. 5.14 and Eq. 5.15 as shown below.

$$N_{K \times K|L} = \frac{(K^2 - S_K) A_m L - A_K A_d}{2 \left(\frac{S_K}{K} - 1 \right) A_r} \quad (5.22)$$

5.3 Summary

Eq. 5.22 suggests that increasing the number of MAC units, L , in each subfilter increases the range of filter lengths over which the $K \times K | L$ FFA structure has a lower area than the $DF | K^2 L$ structure. This can be attributed to the fact that, with increasing levels of parallelism, the relative contribution of the coefficient/data memory area to the total area is reduced. Fig. 5.11 plots the filter length, $N_{K \times K | L}$, below which a FFA based time-shared structure can be implemented with a lower area cost than an equivalent time-shared direct form filter with an identical amount of timing slack. Fig. 5.11, clearly demonstrates that increasing the levels of parallelism, L , in the subfilters, results in the FFA based time-shared filters being more area efficient than time-shared direct form filter over a wider range of filter lengths.

Figure 5.11: $N_{K \times K | L}$ vs. L plot for different FFA structures.

5.3 Summary

In this chapter, we investigated the role of parallelism as a power reductions strategy for time-shared FIR implementations in nanoscale CMOS. It was shown that, even though parallelism incurs an area penalty, the timing slacks obtained by increasing the level of parallelism can be used for reducing all the major nanoscale CMOS power consumption components. A class of time-shared FIR filters based on FFAs was introduced, which was shown to trade area for lower operating frequency more efficiently than traditional folded direct form, up to certain filter lengths.

5.3 Summary

In a multistage decimation filter, a bulk of the potential aliasing components are already removed by the stages prior to the last stage. Hence the last stage filter typically does not have a stringent attenuation requirement, and operates at a low sampling rate. Hence the filter lengths encountered in the last stage are not very high. The area of lower order filters is dominated by the datapath area, and in this region, the proposed class of time-shared FFA structures are more efficient than traditional time-shared filters. The proposed structure exploits algorithmic strength reduction and parallelism for reducing the total power consumption. Traditional strategies for reducing the switching activity in time-shared filters like coefficient reordering, coefficient segmentation, etc. can be used in the temporal style FIR subfilters for further reducing the power consumption [124, 150, 151, 152].

6

Conclusion and Future work

In this chapter we give a brief overview of the contributions and conclusions of the work presented in this thesis as well as some of the directions we have identified for the future work.

6.1 Conclusion

The requirement of flexibility in emerging communication paradigms presents new design challenges in the implementation of HW accelerators, which perform the computationally intensive kernels in the digital baseband of energy constrained mobile terminals. Traditionally, area and power consumption have been the main metrics used for evaluating the performance of a HW accelerator. Incorporating flexibility precludes the use of many high level synthesis behavioral optimizations which make a single mode accelerator area and power efficient. In addition to the area and power penalty, a flexible accelerator may also need a large amount of reconfiguration data to parameterize it for a new mode. A long runtime parametrization time reduces the seamlessness with which the radio can switch between multiple standards in a heterogeneous wireless network, or switch to a new spectrum hole in an opportunistic spectrum access scenario. In this thesis, we have proposed a framework for the design and implementation of accelerators for the computationally intensive channelization function in the digital front-end of a flexible radio handset.

Firstly, we have introduced a notion of the flexibility penalty, which refers to the area, power and reconfiguration latency overheads incurred when flexibility is incorporated

into an accelerator. We have analyzed the factors affecting the area, power penalties in nanoscale CMOS technologies, as well as the practical repercussions of these overheads in a flexible mobile handset. All the components of the flexibility penalty are closely related to the granularity of hardware reuse. Reusing hardware at coarser levels of granularity across multiple modes reduces the area overheads, provides greater opportunities for using behavioral optimizations to reduce the power consumption, and reduces the parametrization overhead. The functionally different channelization tasks of interferer attenuation, channel selection, sample rate conversion and pulse shaping can be reduced to a set of filtering and decimation operations. We have shown that in most of the existing flexible filter accelerator paradigms, the granularity of hardware reuse is limited to coarse grained datapath operators like multipliers, adders and registers. Their dynamic power consumption is significantly higher than accelerators optimized for a single mode. For instance, the total computational load of the channelization accelerator for a standard with a wideband channel like WiMAX is about 5 billion MMAC/s. Performing these operations using generic multipliers incurs a very high dynamic power consumption penalty. A brute force approach for supporting flexibility would be to implement multiple accelerators in parallel (referred to as the Velcro style accelerator), each of them optimized for single standard by using constant-coefficient optimizations. This has the advantage of having a low reconfiguration overhead compared to other accelerator styles, since it provides only the limited flexibility to switch between a fixed set of modes. Its dynamic power consumption is also comparable to that of a single mode accelerator, but such a design suffers from a significant area penalty. Leakage power is becoming increasingly dominant in nanoscale CMOS technologies and is strongly correlated to the area. Thus, reducing the switching activity, area and the reconfiguration latency beyond what is achievable using traditional flexible filter accelerator paradigms requires the hardware functional units to be reused at coarser levels of granularity.

Our second major contribution addresses the problem of finding reusable common operators at a coarser granularity than datapath operators. Performing the decimation and filtering tasks by a single filter stage results in a very long filter. The decimation load is typically split into multiple factors and the required decimations performed by a multistage decimation filter. The filter complexity progressively increases in a multistage decimation chain. Hence, lower order filters operate at a high sampling rate,

while higher order filters operate at a lower sample rate. The filter stages represent a much coarser level of granularity than datapath operators, but their reusability is limited by their dependence on standard specific band-edge and stopband attenuation parameters. It was shown that factorizing arbitrary rational or integral decimation factors into a fixed regular form allowed the stages prior to the last stage to be made independent of standard dependent parameters. These stages can make use of constant-coefficient filter optimizations for reducing the area and power consumption. We have demonstrated the practical application of the multistandard channelization accelerator design framework for 4 standards: GSM, WCDMA, WiMAX and IEEE 802.11a. The proposed design is also in-field upgradeable for supporting a new standard. The high degree of reusability reduces the total area when compared to a Velcro style accelerator. The area of the proposed multistandard design is 65% lower than a Velcro style channelization accelerator for the above 4 standards. Offloading a bulk of the computational load from generic MAC units to hardwired reusable filter stages also reduces the total switching activity and dynamic power consumption. In the proposed structure, only computations in the last stage programmable filter are performed by generic MAC units. Among the 4 standards considered here, the channelization accelerator has the highest computational load in the WiMAX mode. Performing the above computations using generic MAC units incurs a dynamic power consumption of 830 mW. The proposed structure incurs a dynamic power consumption of only 28.4 mW in the WiMAX mode. Thus the proposed structure provides a better area vs power tradeoff than traditional flexible filter accelerators.

While the switching activity in the proposed structure is significantly lower than a generic filter coprocessor, the dynamic power consumption is higher than the corresponding single mode accelerator for any particular mode. This stems from the fact that generic MAC units are required in the last stage filter. Our third contribution addresses the problem of reducing the power consumption of the programmable filter stage. We have shown that parallelism as an architectural transformation can be used for reducing all the major nanoscale CMOS power consumption components, since the parallelism-induced timing slacks allow more room for scaling the threshold and supply voltages. The achievable power savings with parallelism come at the expense of increased functional units. This in turn increase the total physical capacitance and the total gate width, increasing the dynamic and leakage power consumption components

respectively. In this thesis we have proposed a class of time-shared programmable filters that exploit the algorithmic strength property of fast-filter algorithms. With increasing levels of parallelism, the proposed structures have a lower area and total power consumption than direct-form based time-shared filters operating at the same supply and threshold voltage.

6.2 Future Work

6.2.1 Optimum Supply/Threshold Voltage Scaling for Minimizing the Total Power Consumption

In the channelization accelerator structure proposed in this thesis, the stages prior to the last stage are implemented in spatial style, while the last stage is implemented in a temporal fashion. The main limit to supply/threshold voltage scaling is determined by the timing slacks available in the time-multiplexed MAC units in the last stage programmable filter. We have shown that the available timing slacks can be increased by increasing the degree of parallelism by adding more MAC units. The proposed class of FFA based time-shared filters offer greater timing slacks than a direct form based time-shared filter with an identical number of MAC units. The available slack in the MAC units of the last stage filter is a function of the filter length of the last filter, the order of the FFA, the degree of parallelism in the FFA subfilters, and the input sample rate. The filter length and the input sample rate parameters can vary across different modes, resulting in different cycle-periods and varying amount of slack. For any given cycle period, there are multiple (V_{DD}, V_{th}) duplets that result in the MAC delay being equal to the cycle period, thus using up the entire slack. The dynamic and gate leakage power consumption components have a strong dependence on V_{DD} , while the subthreshold leakage power consumption component has a stronger dependence on the V_{th} . The total power consumption of the accelerator can be modeled as follows:

$$P_{tot} = P_{gate}(V_{DD}, W_{total}) + P_{sub}(V_{DD}, V_{th}, W_{total}) + \sum_{i=1}^{N_{cd}} P_{dyn}(V_{DD}, C_{eff}^i, f_i) \quad (6.1)$$

where W_{total} represents the total gate width, N_{cd} represents the total number of clock domains in the channelization accelerator, while C_{eff}^i and f_i represent the effective capacitance and the clock frequency in each of the clock domains.

Points on the (V_{DD}, V_{th}) locus for a given cycle period, lying in the low V_{DD} , high V_{th} region, have a lower dynamic and gate leakage power at the expense of higher subthreshold leakage power, while the points lying in the high V_{DD} , low V_{th} region have a lower subthreshold leakage power at the expense of a higher dynamic and gate leakage power. The optimum (V_{DD}, V_{th}) point which minimizes the total power consumption depends on the relative weight of the different power consumption components. In any given mode, all the parameters in Eq. 6.1 other than the operating voltages are fixed. Hence the task of the finding the optimum (V_{DD}, V_{th}) point can be treated as a constrained optimization, in which the term P_{tot} is minimized subject to the constraint that the MAC delay is equal to the cycle period of a particular mode.

Solving the above optimization problem and evaluating the feasibility of supply/threshold voltage scaling as the nominal V_{DD} values are slowly pushed towards the minimum permissible V_{th} in nanoscale CMOS technologies will be investigated as a part of the future work.

6.2.2 Flexible and Reusable Common Operators in Frequency-domain Implementation of the Channelization Function

In this thesis, we showed that the channelization tasks can be reduced to a set of filtering and decimation operations. We also showed that the filter stages in a multistage decimation filter which performs the above tasks, can be visualized as a reusable common operator, which can be hardwired and reused across multiple modes under certain conditions. So far we have limited our search for common reusable hardware blocks to a time domain implementation of the filtering function. One of the important limitation of the strategies proposed in this thesis is that they are applicable only for hardware sharing and reusing within the channelization function for different modes. The hardwired filter stages cannot be reused by any other function in the SDR baseband. Reusing hardware at coarser levels of granularity provides more room for behavioral optimizations, but reduces the generality and reusability across different functions. A scalable SDR baseband design paradigm also needs to maximize the sharing of common operators across different baseband functions.

Filtering, interpolation and decimation by integer factors and fractional sample rate conversion have all been previously demonstrated in the frequency domain [159, 160, 161, 162]. A frequency domain implementation of the above tasks makes extensive use

of the fast fourier transform (FFT) and the inverse fast fourier transform (IFFT) in the complex field. A number of computationally intensive baseband functions like channel equalization, channel estimation, multicarrier demodulation etc can also be performed in the frequency domain using the FFT and IFFT transforms [163]. A butterfly structure based reconfigurable common operators that can be used as a building block for FFTs and a Galois Field frequency domain implementation of RS encoder/decoder has been proposed in [164], while a similar common operator that can be used for FFTs and Viterbi decoders has been proposed in [165]. FFT based common operators have a high potential for building a scalable and parameterizable SDR baseband.

One of the goals, which we are targeting is to design flexible and reusable FFT based common operators that can be reused both for the channelization function as well as the other computationally intensive baseband kernels. Hardwired FIR filters have very mature constant-coefficient based optimizations which result in significant reduction in the computational complexity. Frequency domain implementations preclude the use of these optimizations. The trade-off between the increase in computational complexity and the reusability when the proposed common operators are used in the SDR baseband will be analyzed as a part of the future work.

Bibliography

- [1] L. Van der Perre, J. Craninckx, and A. Dejonghe, Green Software Defined Radios, *Springer*, pp.1-12, 2009.
- [2] M. Etelper, J.P. Soininen, "4G Mobile Terminal Architecture", *VTT Technical Report*, Nov. Oct, 2007.
- [3] S. Huang, et.al., "Opportunistic spectrum access in cognitive radio networks", in *Proc IEEE INFOCOM*, pp. 1427-1435, 2008.
- [4] S. Haykin, "Cognitive radio: brain-empowered wireless communications," *IEEE J Select. Areas Commun*, vol 23, pp. 201-220, February 2005.
- [5] J. Mitola, "The Software Radio Architecture," *IEEE Commun. Mag.*, vol. 33, no. 5, pp. 26-69, 1995.
- [6] Jeffrey H. Reed, Software Radio: A Modern Approach to Radio Engineering, pp. 2-4, Prentice Hall, 2002.
- [7] A. A. Abidi, "The path to the software-defined radio receiver," *IEEE J.Solid-State Circuits*, vol. 42, no. 5, pp. 954-966, May 2007.
- [8] National Semiconductor, "ADC12D1800- 12-Bit, Single 3.6 GSPS Ultra High-Speed ADC," App. Note, Feb. 2011.
- [9] J.M. Rabaey, "Silicon platforms for the next generation wireless systems - what role does reconfigurable hardware play?," *Proceedings Springer-Verlag*, pp.277-85, 2000.

BIBLIOGRAPHY

- [10] B. Bougard, et al., “A coarse-grained array based baseband processor for 100Mbps+ software defined radio,” in *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE)*, Munich, Germany, pp. 716721, 2008.
- [11] G. Cichon, et al., “Synchronous transfer architecture,” in *Proceedings of the 4th International Workshop on Systems, Architectures, Modeling, and Simulation (SAMOS04)*, pp. 126130, 2004.
- [12] D. Liu, et al., “Bridging dream and reality: programmable baseband processors for software-defined radio,” *IEEE Commun Mag.*, vol. 47, pp. 134140, 2009.
- [13] M.W.Y. Lin, et al., “From SODA to scotch: the evolution of a wireless baseband processor,” in *Proceedings of 41st IEEE/ACM International Symposium on Microarchitecture (MICRO-41)*, Lake Como, Italy, pp. 812, November 2008.
- [14] M. Moudgill, et al., “The Sandblaster 2.0 Architecture and SB3500 Implementation,” in *Proceedings of the Software Defined Radio Technical Forum (SDR Forum 08)*, Washington, DC, October 2008.
- [15] K.V. Berkel, et al., “Vector processing as an enabler for software-defined radio in handheld devices,” *EURASIP Journal on Applied Signal Processing* vol. 2005(1), pp. 2613-2625, 2005.
- [16] P.M. Heysters, G.J.M. Smit, “Mapping of DSP algorithms on the MONTIUM architecture,” *Parallel and Distributed Processing Symposium, 2003. Proceedings International*, pp. 2226, April 2003.
- [17] T. Suzuki, et al., “High Throughput and Low Power Software Defined Radio using Dynamically Reconfigurable Baseband Processor,” *Micro, IEEE*, Preprint, 19 Oct 2011. Online: <http://doi.ieeecomputersociety.org/10.1109/MM.2011.95> (Accessed on 22 Nov, 2011).
- [18] A. Chun, “Key lessons from the scalable communications core: a reconfigurable wireless baseband,” *Communications Magazine, IEEE*, vol.48, no.12, pp.101-109, December 2010.

BIBLIOGRAPHY

- [19] O. Anjum, et al., "State of the art baseband DSP platforms for Software Defined Radio: A survey," *EURASIP Journal on Wireless Communications and Networking*, 2011:5, 2011.
- [20] T. Hentschel, M. Henker and G. P. Fettweis, "The Digital Front-End of Software Radio Terminals", *IEEE Personal Communications*, 6(4):40-46, August 1999.
- [21] R. M. Gray, "Oversampled sigma-delta modulation," *IEEE Trans. Commun.*, COM-35, pp. 481-487, 1987.
- [22] 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 Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 295-304, vol. 24, no.2, Feb. 2005.
- [23] M. Potkonjak, et al., "Multiple Constant Multiplications: Efficient and Versatile Framework and Algorithms for Exploring Common Subexpression Elimination," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 15, no. 2, pp. 151-165, 1996.
- [24] A. G. Dempster and M. D. Macleod, "Use of minimum-adder multiplier blocks in FIR digital filters," *IEEE Trans. Circuits Syst. II, Analog Digit.Signal Process.*, vol. 42, no. 9, pp. 569-577, Sep. 1995.
- [25] Y. Voronenko and M. Pschel, "Multiplierless Multiple Constant Multiplication," *ACM Transactions on Algorithms*, vol. 3, no. 2, 2007.
- [26] C.H Chang, J. Chen and A. P. Vinod, "Information theoretic approach to complexity reduction of FIR filter design," *IEEE Transactions on Circuits and Systems I*, vol. 55, no. 8, pp. 2310-2321, September 2008.
- [27] G. Bellile, O. Dujardin, "Architecture of a programmable FIR filter co-processor, in *Proc. IEEE International Symposium on Circuits and Systems*, pp. 433-436, 1998.
- [28] C.H Stapper, "The effects of wafer to wafer defect density variations on integrated circuit defect and fault distributions," *IBM Journal of Research and Development*, vol.29, no.1, pp.87-97, Jan. 1985.

BIBLIOGRAPHY

- [29] K.Uyttenhove and M.Steyaert, "Speed-power-accuracy trade off in high speed ADC's", *IEEE Trans. Circuits and Systems II*, vol. 4, pp. 247-257, Apr. 2002.
- [30] A. Rusu, A. Borodenkov, M.Ismail and H. Tenhunen, A Triple-Mode Sigma-Delta Modulator for Multi-Standard Wireless Radio Receivers, *Analog Integrated Circuits and Signal Processing*, v.47 n.2, p.113-124, May 2006.
- [31] A. Silva, N.H Horta and J.G. Guilherme, "Reconfigurable multi-mode sigma-delta modulator for 4G mobile terminals", *Integration, The VLSI Journal*, Vol. 42, No. 1, pp. 34 - 46, January, 2009.
- [32] A. Parker and K.K. Parhi, "Low area/power parallel FIR digital filter implementations", *Journal of VLSI Signal Processing*, vol.17, no.1, pp. 75-92, September 1997.
- [33] Z.J. Mou and P. Duhamel, "Short-length FIR filters and their use in fast nonrecursive filtering, " *IEEE Transactions on Signal Processing*, vol. 39, pp. 1322-1332, June 1991.
- [34] B. Premkumar and J. Cai, Principles of Wireless Communications and Data Networks, *Prentice Hall*, 2006.
- [35] U. Ramacher, "Challenges and prospects of SDR for mobile phones," in *Proc. of the 25th International Conference on Computer Design*, pp.215, Oct. 2007.
- [36] A. Nilsson, E. Tell, and D. Liu, "An accelerator architecture for programmable multistandard baseband processors," in *Proc. of WNET2004*, July, 2004.
- [37] P. I. Mak, S. P. U, and R. P. Martins, "Transceiver architecture selection: review, state-of-the-art survey and case study," *IEEE Circuits and Systems Magazine*, vol. 7, no. 2, pp. 6-25, 2007.
- [38] M. Beach, J. MacLeod, and P. Warr, "Radio frequency translation for software defined radios," in *Software Defined Radio: Enabling Technologies*, W. Tuttlebee, Ed., pp. 25-78, John Wiley and Sons, London, UK, 2002.
- [39] A. A. Abidi, "Direct-Conversion Radio Transceivers for Digital Communications," *IEEE Transactions of Solid-State Circuits*, Vol. 30, No. 12, pp.1399-1410, 1995.

BIBLIOGRAPHY

- [40] B. Matinpour, S. Chakraborty, and J. Laskar, "Novel DC-Offset Cancellation Techniques for Even-Harmonic Direct Conversion Receivers," *IEEE Transactions on Microwave Theory and Techniques*, vol. 48, no. 12, pp. 2554-2559, 2000.
- [41] H. Yoshida, H.Tsurumi, and Y.Suzuki, "DC Offset Canceller in a Direct Conversion Receiver for QPSK Signal Reception," *IEEE International Symposium on Personal, Indoor and Mobile Communication*, vol. 3, pp. 1314- 1318, 1998.
- [42] M.Cao, Y.Zheng, and H.K. Garg, "A Novel Algorithm for DC Offset and Flicker Noise Cancellation in Direct Conversion Receivers," *International Conference on Communications and Systems*, pp. 441-445, 2004.
- [43] S.J. Fang, A. Bellaouar, S.T. Lee, and D. J. Allstot, "An Image-Rejection Down-Converter for Low-IF Receivers," *IEEE Transactions on Microwave Theory and Techniques*, vol. 53, no. 2, pp. 478-487, 2005.
- [44] Adiseno, M. Ismail, and H. Olsson, "A Wide-Band RF Front-End for Multiband Multistandard High-Linearity Low-IF Wireless Receivers," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 9, pp. 1162-1168, 2002.
- [45] H. Berndt, R. Richter, and H.J. Jentschel, "A 100 MS/sec, 8thorder Quadrature Sigma-Delta ADC for Complex-IF Signal Digitization in a Wideband-IF Sampling Receiver," *Proceedings of International Conference on ASIC*, vol. 1, pp. 669-672, 2003.
- [46] J. C. Rudell, et.al, "A 1.9-GHz Wide-Band IF Double Conversion CMOS Receiver for Cordless Telephone Applications," *IEEE Journal of Solid-State Circuits*, vol. 32, no. 12, pp. 2071-2088, 1997.
- [47] S. Albrecht, "Sigma-delta based techniques for future multi-standard wireless radios," PhD Thesis, KTH, 2005.
- [48] A. Rusu, B.Dong and M. Ismail, "Putting the "FLEX" in flexible mobile wireless radios - A Wideband Bandpass Sigma-Delta ADC for Software Radios," *IEEE Circuits and Devices Magazine*, vol. 22, no. 6, pp. 24-30, 2006.

BIBLIOGRAPHY

- [49] J. Markus, and G.C.Temes, "An Efficient Delta-Sigma ADC Architecture for Low Oversampling Ratios," *IEEE Trans. Circuits and Sys.-I*, vol. 51, no.1, pp. 63-71, 2004.
- [50] A. Gothenberg and H. Tenhunen, "Low oversampling ratio sigma-delta noise shaper architecture for wideband radio applications," *In Proc. IEEE NORCHIP Conference*, pp. 196-206, 1998.
- [51] O. Shoaie, "Continuous-Time Delta-Sigma A/D Converters for High Speed Applications," PhD Thesis, Carleton University, 1995.
- [52] F.Henkel, et.al, "A 1-MHz-Bandwidth Second-Order Continuous-Time Quadrature Bandpass Sigma-Delta Modulator For Low-IF Radio Receivers," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 12, pp. 1628-1635, 2002.
- [53] M. Lhning, T. Hentschel and G. P. Fettweis, "Digital Down Conversion in Software Radio Terminals," *in Proceedings of the European Signal Processing Conference*, vol. 3, pp. 1517-1520, 2000.
- [54] K. Gentile, "Digital Pulse-Shaping Filter Basics," *Analog Devices, Appl. Note, AN-992*, pp. 1-12.
- [55] M. Bellanger, "On Computational Complexity in Digital Filters," *in Proc. Of the European Conference on Circuit Theory and Design*, pp. 58-63, 1981.
- [56] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*, Prentice Hall, PTR, 1993.
- [57] W. J. Dally, et al., "Energy efficient computing, *IEEE Comput. Mag.*, vol. 41, no. 7, pp. 2732, July 2008.
- [58] M. Gries, K. Ketuzer, H. Meyr and G. Martin, *Building ASIPS: The Mescal Methodology*. New York: Springer, 2005.
- [59] P. Ienne and R. Leupers, *Customizable Embedded Processors: Design Technologies and Applications*. San Francisco, CA: Morgan Kaufman, 2006.
- [60] J. Vankka, *Digital Synthesizers and Transmitters for Software Radio*, Springer-Verlag New York, pp. 239-257, 2005 .

BIBLIOGRAPHY

- [61] L.R. Rabiner, Multirate Digital Signal Processing, Prentice Hall PTR, Upper Saddle River, NJ, 1996.
- [62] Z. Tao and S. Signell, "Multi-standard delta-sigma decimation filter design", in *Proc. of IEEE Asia Pacific Conference on Circuits and Systems*, pp. 1212-1215, 2006.
- [63] J.C. Whitaker, "The Electronics Handbook", CRC Press, 1996.
- [64] T. Hentschel and G. P. Fettweis, "Sample Rate Conversion for Software Radio", in *IEEE Communications Magazine*, vol. 38, no. 8, pp. 142-150, 2000.
- [65] E.B. Hogenauer, "An economical class of digital filters for decimation and interpolation," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-29, no.2, pp.155-162, April 1981.
- [66] Y. Gao, L. Jia, J. Isoaho, and H. Tenhunen, "A comparison design of comb decimators for sigma-delta analog-to-digital converters," in *Analog Integrated Circuits and Signal Processing*, vol.22, pp.51-60, 1999.
- [67] H. Aboushady, Y. Dumonteix, M.-M. Louerat, H. Mehrez, "Efficient polyphase decomposition of comb decimation filters in $\Sigma\Delta$ analog-to-digital converters," in *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, vol.48, no.10, pp.898-903, Oct 2001.
- [68] A. Kwentus, Z. Jiang, and A. Willson Jr., "Application of filter sharpening to cascaded integrator comb decimation filters," in *IEEE Trans. on Signal Processing*, vol. 45, pp. 457-467, February 1997.
- [69] S.B. Chu, C. Burrus, "Multirate filter designs using comb filters," in *Circuits and Systems, IEEE Transactions on*, vol.31, no.11, pp. 913-924, Nov 1984.
- [70] L.L Presti, "Efficient modified-sinc filters for sigma-delta A/D converters," in *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol.47, no.11, pp.1204-1213, Nov. 2000.
- [71] H.J. Oh, S. Kim, G. Choi, Y.H. Lee, "On the use of interpolated second-order polynomials for efficient filter design in programmable downconversion," in *Selected Areas in Communications, IEEE Journal on*, vol.17, no.4, pp.551-560, Apr 1999.

BIBLIOGRAPHY

- [72] G.J. Dolecek, "Novel multiplierless wide-band CIC compensator," *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, pp.2119-2122, 2010.
- [73] C. W. Farrow, "A continuously variable digital delay element," in *Proc. IEEE Int. Symp. Circ. Syst.*, vol. 3, pp. 2641-2645, June 1988.
- [74] A. Tkachenko, "Variable sample rate conversion techniques for the Advanced Receiver," *Interplanetary Network (IPN) Progress Report*, vol. 42-168, Feb. 15, 2007.
- [75] J. Vesma, T. Saramki, "Interpolation filters with arbitrary frequency response for all-digital receivers," in *Proc. IEEE Int. Symp. Circuits Syst.*, pp. 568-571, May 1996.
- [76] S. Mirabbasi, K. Martin, "IIR digital filter for $\Sigma\Delta$ decimation, channel selection, and square-root raised-cosine nyquist filtering," *Solid-State Circuits Conference, 2002. Digest of Technical Papers. ISSCC. 2002 IEEE International*, vol.2, no., pp.96-417, 2002.
- [77] L.A. de B. Naviner, J.-F. Naviner, and E.Colin, "Decimation filter cascade for channel selection in multistandard receivers," *Circuits and Systems, 2003 IEEE 46th Midwest Symposium on*, vol.1, no., pp.245-248 Vol. 1, 30-30 Dec. 2003.
- [78] A. Ghazel, L. Naviner, K. Grati, "On design and implementation of a decimation filter for multistandard wireless transceivers," *Wireless Communications, IEEE Transactions on*, vol.1, no.4, pp. 558-562, Oct 2002.
- [79] K. Vaisanen, M. Renfors, "Efficient digital filters for pulse-shaping and jitter-free frequency error detection and timing recovery," *Signal Processing*, volume 81, no. 4, pp. 829-844, April 2001.
- [80] T. K. Shahana, et.al, "Decimation Filter Design Toolbox for Multi-Standard Wireless Transceivers using MATLAB," *International Journal of Signal Processing*, Volume 5, Number 2, pp.154, Spring 2009.
- [81] C. J. Barrett, "Low-Power Decimation Filter Design for Multi-Standard," University of California, Berkeley, Technical Report No. UCB/ERL M97/88, 1997.

BIBLIOGRAPHY

- [82] T.K. Shahana, "Dual-mode RNS based programmable decimation filter for WCDMA and WLANa," *Circuits and Systems, 2008. ISCAS 2008. IEEE International Symposium on*, pp.952-955, 2008.
- [83] T. Wang; C. Li, "Sample Rate Conversion Technology in Software Defined Radio," *in Proc. of the Canadian Conference on Electrical and Computer Engineering*, pp.1355-1358, May 2006.
- [84] N. Khouja, et al., "Low power design of a decimation filter in multi-stand standard receiver," *Design and Technology of Integrated Systems in Nanoscale Era, 2008. DTIS 2008. 3rd International Conference on*, pp.1-5, 2008.
- [85] C. Martelli, et al., "A 50mW HSDPA Baseband Receiver ASIC with Multimode Digital Front-End," *Solid-State Circuits Conference, 2007. ISSCC 2007. Digest of Technical Papers. IEEE International*, pp.260-601, 2007.
- [86] M. Laddomada, D.E. Troncoso, G.D. Jovanovic, "Design of Multiplierless Decimation Filters Using an Extended Search of Cyclotomic Polynomials," *Circuits and Systems II: Express Briefs, IEEE Transactions on*, vol.58, no.2, pp.115-119, Feb. 2011.
- [87] M.A. Siddiqi, N. Samad, S.Masud, F. Sheikh, "FPGA-based Implementation of Efficient Sample Rate Conversion for Software Defined Radios," *Computer and Information Technology (CIT), 2010 IEEE 10th International Conference on*, pp.2387-2390, 2010.
- [88] F. Sheikh, S. Masud, "Efficient sample rate conversion for multi-standard software defined radios, *IEEE Int. Conf. on Acoustics, Speech and Signal Processing, HI*, pp. II-329 II-332, April 2007
- [89] H. Johansson and O. Gustafsson, "Linear-phase FIR interpolation, decimation, and Mth-band filters utilizing the Farrow structure, *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 52, no. 10, pp. 2197-2207, Oct. 2005.
- [90] H. Johansson, "Farrow-Structure-Based Reconfigurable Bandpass Linear-Phase FIR Filters for Integer Sampling Rate Conversion," *Circuits and Systems II: Express Briefs, IEEE Transactions on*, vol.58, no.1, pp.46-50, Jan. 2011.

BIBLIOGRAPHY

- [91] G. D. Micheli, *Synthesis and Optimization of Digital Circuits*. McGraw-Hill, 1994.
- [92] M. Lundstorm, "Moore's law forever, *Science*," Vol. 299. no. 5604, pp. 210 - 211, 2003.
- [93] B. Nikolic, "Design in the power-limited scaling regime," *IEEE Transactions on Electron Devices*, vol.55, no.1, pp. 71-83, January 2008.
- [94] K. Roy, et al., "Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits," *Proceedings of the IEEE*, vol.91, no.2, pp. 305-327, February 2003.
- [95] N. S. Kim, et al., "Leakage current: Moore's law meets static power," *Computer*, vol. 36, no. 12, pp. 68-75, December 2003.
- [96] H. Jeon, Y.B. Kim, "A novel technique to minimize standby leakage power in nanoscale CMOS VLSI", *IEEE International Instrumentation and Measurement Technology Conference*, pp. 1372-1375, May5-7, 2009.
- [97] M. Drazdziulis, "A gate leakage reduction strategy for future CMOS circuits," in *Proc.Eur. Solid-State Circuits, Conf.*, pp.317-320, September 2003.
- [98] S.P. Mohanty, et al., "Analytical modeling and reduction of direct tunneling current during behavioral synthesis of nanometer CMOS circuits," in *Proceedings of the 14th ACM/IEEE International Workshop on Logic and Synthesis (IWLS)*, pp.249-256, 2005.
- [99] E. Kougianos and S.P. Mohanty. "Impact of gate-oxide tunneling on mixed-signal design and simulation of a nano-CMOS VCO," *Microelectronics J.*, vol. 40, no.1, pp. 95-103, 2009.
- [100] A.P. Chandrakasan and R.W. Brodersen, "Low power CMOS digital design," *Norwell, MA: Kluwer*, 1996.
- [101] C.H Stapper, "The effects of wafer to wafer defect density variations on integrated circuit defect and fault distributions," *IBM Journal of Research and Development*, vol.29, no.1, pp.87-97, Jan. 1985.

BIBLIOGRAPHY

- [102] Y. Shin, S. Heo, H. Kim, and J. Choi, "Supply switching with ground collapse: Simultaneous control of subthreshold and gate leakage current in nanometer-scale CMOS circuits, *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 15, no. 7, pp. 758-766, Jul. 2007.
- [103] W.Y. Lee, I. Akyildiz, "Optimal spectrum sensing framework for cognitive radio networks," *IEEE Transactions on Wireless Communications*, vol. 7, no. 10, pp. 3845-3857, 2008.
- [104] D. Cabric, S.M. Mishra, and R.W. Brodersen, "Implementation issues in spectrum sensing for cognitive radio," in *Proc. Asilomar Conf. on Signals, Syst., and Comput.*, pp. 772-776, 2004.
- [105] G. Ganesan, Y. Li, "Cooperative spectrum sensing in cognitive radio networks," in *Proc. of IEEE DySPAN*, pp. 137-143, 2005.
- [106] A. Ghasemi, E.S. Sousa, "Collaborative spectrum sensing for opportunistic access in fading environments," in *Proc. of IEEE DySPAN*, pp. 131-136, 2005.
- [107] C.T. Chou, et.al., "What and how much to gain by spectral agility?," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 3, pp. 576-588, 2007.
- [108] Caaliph, et. al., "Synthesis of Multimode digital signal processing systems," in *Proc. Second NASA/ESA Conference on Adaptive Hardware and Systems*, pp. 318-325, 2007.
- [109] L. Chiou, et. al., "Synthesis of application-specific highly-efficient multi-mode systems for low-power applications. Trans. On Embedded Computing Systems", vol. 4, no. 1, pp. 168-188, 2005.
- [110] V. Kumar and J. Lach, "Highly flexible multimode digital signal processing systems using adaptable components and controllers," *EURASIP Journal on Applied Signal Processing*, 2006.
- [111] N. Moreano, G. Araujo, and C.C de Souza, "CDFG merging for reconfigurable architectures," Tech. Rep. IC-03-18, Institute of Computing, University of Campinas SP, Brazil, 2003.

BIBLIOGRAPHY

- [112] B. L. Gal, et.al., "HLS design flow for the synthesis of multimode systems under multiple constraints," in *Proc. IEEE International Conference on Electronics, Circuits and Systems*, pp.314-317, 2007.
- [113] P. Tummeltshammer, J. C. Hoe and M. Pschel, "Time-Multiplexed Multiple Constant Multiplication," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 9, pp. 1551-1563, 2007.
- [114] R. Hartenstein, Basics of Reconfigurable Computing: Embedded Computing - A Low Power Perspective, Springer-Verlag, 2007.
- [115] K. Chapman, "Fast Integer Multipliers fit in FPGA's," *EDN 1993 Design Idea Winner*, *EDN*, May 12th, 1994.
- [116] P. Lysaght, J. Dunlop, "Dynamic reconfiguration of field programmable gate arrays," in *Proc. Int. Workshop on Field Programmable Logic and Applications*, pp.82-94, 1993.
- [117] Xilinx Inc, Virtex-4 Configuration Guide, UG071,v1.1, 2004.
- [118] J. Delorme et. al., "New OPBHWICAP interface for real-time Partial reconfiguration of FPGA," in *Proc. of the International Conference on ReConFigurable Computing and FPGAs*, pp. 386-391, 2009.
- [119] K.G. Smitha, A.P. Vinod, "A Reconfigurable Low Complexity Architecture For Channel adaptation in Cognitive Radio, "in *Proc. of IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, pp.1-5, 2008.
- [120] X. Chong, et. al., "Order-Configurable Programmable Power-Efficient FIR Filters," in *Proc. Third International Conference on High-Performance Computing*, pp. 357-361, 1996.
- [121] S. Ramanathan, et.al., "Reconfigurable Filter Coprocessor Architecture for DSP Applications," *The Journal of VLSI Signal Processing*, vol. 26, no. 3, pp. 333-359, 2000.
- [122] Intersil, Serial I/O Filter, Datasheet, HSP43124, 2007.
- [123] Synopsys Inc, DW_fir, High-Speed Digital FIR Filter, Datasheet, 2009.

BIBLIOGRAPHY

- [124] A. T. Erdogan, T. Arslan, "Low Power FIR Filter Implementations Based on Coefficient Ordering Algorithm," in *Proc. of IEEE Computer Society Annual Symposium on VLSI: Emerging Trends in VLSI Systems Design*, pp. 226-228, 2004.
- [125] H. Samueli, "An Improved Search Algorithm for the Design of Multiplierless FIR Filters with Power-of-Two Coefficients," *IEEE Trans. CAS*, vol. 36, pp. 1044-1047, 1989.
- [126] K.H. Chen and T.D. Chiueh, "A Low-Power Digit-Based Reconfigurable FIR Filter," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 53, no. 8, pp. 617-621, 2006.
- [127] K.Y. Khoo et.al., "A programmable FIR digital filter using CSD coefficients," *IEEE J. Solid-State Circuits*, vol. 31, no. 6, pp. 869-874, 1996.
- [128] T. Zhangwen et.al., "A high-speed, programmable CSD coefficient FIR Filter," in *Proc. 4th Int. Conf. ASIC*, pp.397-400, 2001.
- [129] T. Rissa, et. al., "Adaptive FIR filter architectures for run-time reconfigurable FPGAs", in *Proc. IEEE International Conference on Field-Programmable Technology*, pp. 52-59, 2002.
- [130] 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 and Systems*, vol. 27, no.2, pp.217-229, 2008.
- [131] R. Mahesh, A. P. Vinod, "New Reconfigurable Architectures for Implementing FIR Filters With Low Complexity," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 29, no.2, pp. 275-288, 2010.
- [132] D.D. Gajski, et. al., "Essential issues for IP reuse," in *Proc. of Asia and South Pacific Design Automation Conference*, pp.37-42, 2000.
- [133] Z. Junchao, et. al., "Parameterized IP core design," in *Proc. of the 4th International Conference on ASIC*, pp.744-747, 2001.

BIBLIOGRAPHY

- [134] V. Rodriguez, C. Moy and J. Palicot, “Install or invoke ?: The optimal tradeoff between performance and cost in the design of multi-standard reconfigurable radios,” *Wiley InterScience.Wireless Communications and Mobile Computing Journal*, vol. 7, no. 9, pp.1143–1156, 2007.
- [135] *Wireless Communications Specifications*, RF Cafe. [Online]. Available: <http://www.rfcafe.com/references/electrical/wireless-comm-specs-new.htm> (Accessed on 24 June, 2010).
- [136] T. Hentschel and G. Fettweis , “Continuous-time digital filters for sample-rate conversion in reconfigurable radio terminals,” *Frequenz*, vol. 55, pp. 185-188, 2001.
- [137] J.G Proakis, D.G. Manolakis, “Digital Signal Processing,” *Pearson Education*, 2004.
- [138] M. Unser, M. Eden, “FIR Approximations of Inverse Filters and Perfect Reconstruction Filter Banks,” *Signal Processing*, vol. 36, no. 2, pp. 163-174, March 1994.
- [139] A. T. Erdogan, T. Arslan, “Low Power FIR Filter Implementations Based on Coefficient Ordering Algorithm,” *IEEE Computer Society Annual Symposium on VLSI: Emerging Trends in VLSI Systems Design*, pp.226, 2004.
- [140] N. Michael, C. Moy, A. P. Vinod and J. Palicot, “Area-Power tradeoffs for flexible filtering in green radios,” *Journal of Communications and Networks*, vol.12, no.2, pp. 158-167, April 2010.
- [141] N. Michael, A. P. Vinod, C. Moy and J. Palicot, “Area-efficient time-shared FIR filters in nanoscale CMOS,” *Proceedings of IEEE International Conference on Green Circuits and Systems*, pp. 54-59, June 2010.
- [142] H. Tarn, E. Hemphill, and D. Hawke, “3GPP LTE Digital Front End Reference Design,” *Xilinx Application Note XAPP1123 (v1.0)*, Oct 2008.
- [143] Texas Instruments, ‘TMS320C6472 Fixed-Point Digital Signal Processor’, Datasheet , June 2011. Available Online : www.ti.com/lit/ds/sprs612g/sprs612g.pdf (Accessed on 24/11/2011)

BIBLIOGRAPHY

- [144] L. Truong, “Low power consumption and a competitive price tag make the six-core TMS320C6472 ideal for high-performance applications”, Texas Instruments Whitepaper, Oct 2009. Available Online: www.ti.com/lit/wp/spry130/spry130.pdf (Accessed on 24/11/2011)
- [145] P. M. Heysters, “Coarse-grained reconfigurable processors,” Ctit Ph.D. thesis series no. 04-66, University of Twente, 2004. ISBN: 90-365-2076-2.
- [146] T.T Hoang, M. Sjalander, P. Larsson-Edefors, “A High-Speed, Energy-Efficient Two-Cycle Multiply-Accumulate (MAC) Architecture and Its Application to a Double-Throughput MAC Unit,” *Circuits and Systems I: Regular Papers, IEEE Transactions on* , vol.57, no.12, pp.3073-3081, Dec. 2010.
- [147] R. Achenbach, et.al., “A digitally temperature-compensated crystal oscillator,” *Solid-State Circuits, IEEE Journal of*, vol.35, no.10, pp.1502-1506, Oct 2000.
- [148] C. Barret (Ed.), “Fractional/integer-N PLL basics, Texas Instruments Wireless Communication Business Unit, Technical Brief SWRA029, August 1999. [Online]. Available: <http://focus.ti.com/docs/apps/catalog/resources/appnoteabstract.jhtml?abstractName=swra029>
- [149] B.G Goldberg, Digital Frequency Synthesis Demystified, LLH Technology Publishing, pp. 82-88, 1999.
- [150] V. Sundararajan and K.K. Parhi, “Synthesis of Low Power Folded Programmable Coefficient FIR Digital Filters,” in *Proc. of IEEE Asia Pacific Design Automation Conference*, pp. 153-156, 2000.
- [151] T. Arslan and A. T. Erdogan, “Data block processing for low power implementation of direct form FIR filters on single multiplier CMOS DSPs,” *IEEE International Symposium on Circuit and Systems*, vol.5, pp. 441-444, 1998.
- [152] A. T. Erdogan and T. Arslan, “A coefficient segmentation algorithm for low power implementation of FIR filters,” *IEEE International Symposium on Circuits and Systems*, pp. 359-362, 1999.

BIBLIOGRAPHY

- [153] D. Liu, C. Svensson, "Trading speed for low power by choice of supply and threshold voltages," *Solid-State Circuits, IEEE Journal of*, vol. 28, no. 1, pp. 10-17, Jan 1993.
- [154] K. A. Bowman, et al., "A physical alpha-power law MOSFET model", in *Proceedings of International Symposium on Low Power Electronics and Design*, pp. 218-222, 1999.
- [155] S. Narendra, et al., "Forward body bias for microprocessors in 130nm technology generation and beyond," *IEEE J. Solid-State Circuits*, vol.38, pp. 696-701, 2003.
- [156] N. Sirisantana, et al., "High-performance low-power CMOS circuits using multiple channel length and multiple oxide thickness," in *Proceedings IEEE International Conference on Computer Design*, pp. 227-232, 2000.
- [157] A. Keshavarzi, et al., "Effectiveness of reverse body bias for leakage control in scaled dual Vt CMOS IC," in *Proceedings of the International Symposium on Low Power Electronics and Design*, pp. 207-212, 2001.
- [158] M. Mehendale, S.D. Sherlekar, G. Venkatesh, "Low power realization of FIR filters using multirate architectures", *9th International Conference on VLSI Design: VLSI in Mobile Communication*, pp.370-375,1996.
- [159] M. Ojail, et al., "A Frequency Domain FIR Filter Implementation Method for 3G Communication Systems," *Digital Telecommunications, 2008. ICDT '08. The Third International Conference on*, pp.1-5, 2008.
- [160] B. Guoan, S.K. Mitra, "Sampling Rate Conversion in the Frequency Domain [DSP Tips and Tricks], *Signal Processing Magazine, IEEE*, vol. 28, no. 3, pp. 140-144, May 2011.
- [161] B. Gold and C. Rader, *Digital Processing of Signals*, New York: McGraw-Hill, 1969.
- [162] M. Yeh, J. Melsa, and D. Cohn, "A direct FFT scheme for interpolation, decimation and amplitude modulation," in *Proc. 16th Asilomar Conf. Circuits, Syst., Computers*, pp. 437-441, 1982.

BIBLIOGRAPHY

- [163] J. Palicot, C. Roland, "FFT: a basic function for a reconfigurable receiver," *Telecommunications, 2003. ICT 2003. 10th International Conference on*, vol.1, no., pp. 898- 902 vol.1, 23 Feb.-1 March 2003.
- [164] A. A. Ghouwayel, Y. Lout, and J. Palicot, "A reconfigurable architecture for the FFT operator in a software radio context," *in Proceedings of IEEE International Symposium on Circuits and Systems*, pp. 181-184, May 2006.
- [165] M. Naoues, L. Alaus, D. Noguet, "A Common Operator for FFT and Viterbi Algorithms," *Digital System Design: Architectures, Methods and Tools (DSD), 2010 13th Euromicro Conference on*, pp.309-313, 1-3 Sept. 2010.
- [166] M. Pschel, et al., "SPIRAL: Code Generation for DSP Transforms," *in Proc. of the IEEE special issue on "Program Generation, Optimization, and Adaptation,"* vol. 93, no. 2, pp. 232-275, 2005.