UNIVERSITY OF CALIFORNIA

Los Angeles

Energy-Efficient VLSI Architectures for Next-Generation Software-Defined and Cognitive Radios

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Electrical Engineering

by

Fang-Li Yuan

2014

ABSTRACT OF THE DISSERTATION

Energy-Efficient VLSI Architectures for Next-Generation Software-Defined and Cognitive Radios

by

Fang-Li Yuan

Doctor of Philosophy in Electrical Engineering

University of California, Los Angeles, 2014

Professor Dejan Marković, Chair

Dedicated radio hardware is no longer promising as it was in the past. Today, the support of diverse standards dictates more flexible solutions. Software-defined radio (SDR) provides the flexibility by replacing dedicated blocks (i.e. ASICs) with more general processors to adapt to various functions, standards and even allow mutable design changes. However, such replacement generally incurs significant efficiency loss in circuits, hindering its feasibility for energy-constrained devices. The capability of dynamic and blind spectrum analysis, as featured in the cognitive radio (CR) technology, makes chip implementation even more challenging.

This work discusses several design techniques to achieve near-ASIC energy efficiency while providing the flexibility required by software-defined and cognitive radios. The algorithm-architecture co-design is used to determine domain-specific dataflow

structures to achieve the right balance between energy efficiency and flexibility. The flexible instruction-set-architecture (ISA), the multi-scale interconnects, and the multi-core dynamic scheduling are also proposed to reduce the energy overhead. We demonstrate these concepts on two real-time blind classification chips for CR spectrum analysis, as well as a 16-core processor for baseband SDR signal processing. The blind classifier achieves a 59$\times$ lower energy compared to an exhaustive method, while the 16-core SDR processor shows $>$2.4$\times$ higher energy efficiency than state-of-the-art communication processors and closes the gap with functionally-equivalent ASICs to within 2.6$\times$. These techniques not only enable energy-efficient and flexible radio implementation, but can also be applied to other domains of computing.

The dissertation of Fang-Li Yuan is approved.

Gregory P. Carman

Gregory J. Pottie

Dejan Marković, Committee Chair

University of California, Los Angeles

2014

To my dear brother and parents.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

My sincere appreciation goes to my advisor, Professor Dejan Marković. His excellence in academic knowledge and true wisdom in research foresight have always enlightened me to develop many ideas throughout my Ph.D. journey. Dejan's constant support in funding acquisition illustrates his great endeavor in giving the best economic treatment he could for his students to fully focus on research. His caring friendship eliminates the generally hard barrier between an advisor and a student, thus giving me freedom to share everything with him, whether it's about research or our daily life. I especially thank Dejan for his faith in my research and his tolerance of my weaknesses. His patient, trustful, and encouraging support has made me a better scholar and a better person. I also thank my committee members, Professor Kung Yao, Professor Gregory J. Pottie, and Professor Gregory P. Carman, to provide thoughtful evaluation of my research proposal and review of my dissertation.

I have been blessed to work with my labmates in UCLA Parallel Data Architecture (PDA) Group. Foremost gratitude goes to Professor Chia-Hsiang Yang (now with NCTU, Taiwan), Dr. Tsung-Han Yu and Dr. Cheng C. Wang. Chia-Hsiang's unselfishness in sharing his wisdom and experiences helped me a lot to get through pressures during chip designs and paper writing. I thank Tsung-Han, my senior at NTU EE by one year, for being my best friend, my great roommate and for his long-term support since our first meeting in 2002. Cheng was the first people I met in the lab. We sat in the

same cubicle for five years, and have always been the closest partner through the FPGA project and many tapeouts that involved countless joy and pain. I also thank other PDA members – Dr. Chaitali Biswas, Dr. Sarah Gibson, Dr. Vaibhav Karkare, Dr. Rashmi Nanda, Hariprasad Chandrakumar, Sina Basir-Kazeruni, Vahagn Hokhikyan, Richard Dorrance, Fengbo Ren, Dejan Rozgic and Yuta Toriyama – for their precious discussions and friendship. I thank Henry Chen for his help with chip testing.

The DARPA/CLASIC project created many opportunities to further develop my work in collaboration with UCLA Cognitive Reconfigurable Embedded Systems (CORES) Lab and UMN Analog Research Lab. I thank Paulo Isagani Urriza, Dr. Tsung-Han Yu, Dr. Eric Rebeiz and Professor Danijela Čabrić for their invaluable insights that finally lead to some good publications. I would also like to thank the EE Department for continued assistance regarding server maintenance, stipend/reimbursement management, event setup, and many other things. Sincere gratitude goes to Kyle Jung and Deeona Columbia, for their enthusiasm and kindness in problem solving.

This work is supported by the FRCP/C2S2 program, the DARPA/CLASIC program, and the Broadcom Fellowship. I thank my fellowship mentors Dr. David Garrett and Dr. Mehdi Hatamian for technical assistance and inspiring guidance from Broadcom. Chip fabrication services are provided by ST Microelectronics, TSMC and IBM.

Surviving graduate school was made much easier with the accompany from many friends in the States. I thank Wen-Tai Huang, Yen-Ting Matt Lin, Po-Wei Chen and Andrew Lin for joyful memories on the golf courses, plus Yen-Ju Chen, Tim Ou, Hao Ho, Yuta Toriyama, Cheng C. Wang, Tsung-Han Yu, Jack Wei, Hung-Li Chang, Kuang-

Hao Su, Insky Chen, Linya Wang and many others for the good time we spent during the snow seasons. I am grateful to Hann Wang and Stephanie Tong for their enthusiastic efforts on the UCLA food club. Matt's two sons always make my day with their cute acts. Thank my roommates Ian Chu, Chun-Hao Liu and Yeh-Jung Hsiao for the lovely moments in 1627 Manning Avenue. Thank Eugene Kung, Yu-Jui Fan and many others for the interactive encouragement about our Ph.D. goals. I proudly appreciate the academic contributions that some NTU EE 106^{th} classmates – Jeffrey Lee, Yu-Hsiu Wu, Chung-You Lou, Tsung-Tse Wu, Chung-Kai Yu, Dr. Jim Sun and Dr. Jay Chien – have made to UCLA EE as Ph.D. candidates/alumni. The occasional gathering still reminds me of the good old time we had in NTU. Keeping in touch with other classmates such as Bernie Yang (in San Diego), Joseph Lin and Edward Kao (in New York) makes me especially happy when I visit those cities. I won't forget the good time when my dear high-school classmates Ping-I Chen, Kuan-Chen Chin and Yuan-Chin Chiou visited me. I owe all of them many wonderful memories.

Special thanks go to Professor Chorng-Kuang Wang, my former advisor during my master studies in NTU. My Ph.D. adventure wouldn't have started without his encouragement and support that eventually fulfilled my dream of studying in UCLA.

Finally, and most importantly, I would like to express my greatest gratitude to my parents Mrs. Li-Chu Liang and Mr. Wan-Liang Yuan, my brother Fang-Chen Yuan, and my girlfriend Ya-Yun Cheng. I thank them for their understanding of my hectic schedule, especially during chip tapeout. This work is a product of their endless love and unconditional support.

VITA

2002–2006 B.S. in Electrical Engineering, National Taiwan University, Taipei, Taiwan.

2006–2008 M.S. in Electronics Engineering, National Taiwan University, Taipei, Taiwan.

PUBLICATIONS

F.-L. Yuan, T.-H. Yu, and D. Marković, "A 500MHz Blind Classification Processor for Cognitive Radios in 40nm CMOS," in *proc. IEEE International Symposium on VLSI Circuits (VLSI)*, paper 8.3, June 2014.

F.-L. Yuan and D. Marković, "A 13.1GOPS/mW 16-Core Processor for Software-Defined Radios in 40nm CMOS," in *proc. IEEE International Symposium on VLSI Circuits (VLSI)*, paper 8.4, June 2014.

C. C. Wang, F.-L. Yuan, T.-H. Yu, and D. Marković, "A Multi-Granularity FPGA with Hierarchical Interconnects for Efficient and Flexible Mobile Computing," in *proc. IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pp. 460-461, Feb. 2014.

E. Rebeiz, F.-L. Yuan, P. Urriza, D. Marković, and D. Čabrić, "Energy-Efficient Processor for Blind Signal Classification in Cognitive Radio Networks," *IEEE Transactions on Circuits and Systems I: Regular Papers (TCAS-I)*, vol. 51, no. 2, pp. 587-599, Feb. 2014.

F.-L. Yuan, C.-H. Yang, and D. Marković, "A Hardware-Efficient VLSI Architecture for Hybrid Sphere-MCMC Detection," in *proc. IEEE Global Telecommunication Conference (GLOBECOM)*, pp. 1-6, Dec. 2011.

C. C. Wang, F.-L. Yuan, H. Chen, and D. Marković, "A 1.1 GOPS/mW FPGA with Hierarchical Interconnect Fabric," in *proc. IEEE International Symposium on VLSI Circuits (VLSI)*, pp. 136-137, June 2011.

CHAPTER 1

Introduction

1.1 Modern Communications: Evolution and Challenges

Modern communication technology plays a pivotal role to help people exchange information efficiently. Ever since the invention of the world's first mobile phone in the 1980s, wireless technology has rapidly proliferated into all aspects of our life, and the demand for high-speed and reliable wireless connectivity is ever increasing. The demand drives the fast evolution of wireless standards. Today's third-generation (3G) and 4G devices are able to process not only voice/text but also real-time video streams, making user experiences better than ever.

However, the fast evolution poses new emphasis on hardware design: *flexibility* and *energy efficiency*.

Flexibility: As shown in Fig. 1.1, diverse radio protocols have been established for a rich selection of uplink/downlink scenarios and communication distance during the last decade. Some of them are finalized (e.g. IEEE 802.11a), but some are still evolving (e.g. LTE). Their parameters, e.g. signal bandwidth, are different depending on target applications such as mobility or communication range. While future systems require

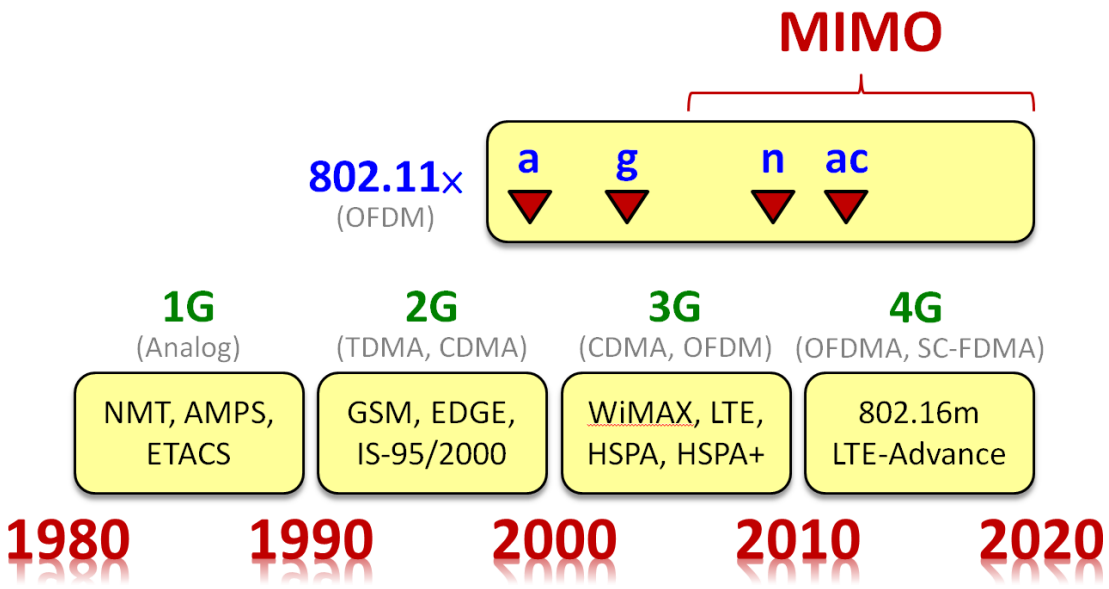


Figure 1.1: The evolution of wireless communication standards.

multi-mode capability to optimize their performance, a flexible digital hardware compatible with existing and evolving standards is highly desired. In addition, flexibility implies the ability to support design changes, thereby reducing the cost of fabrication rework (several million USD in deep-submicron technologies) and shortening the time to market.

Energy efficiency: As seen in Fig. 1.1, the multiple-input multiple-output (MIMO) technology, featuring its high spectral efficiency and space-time-coding capability, has been widely adopted since 2004 to improve the link throughput and/or the transmission robustness.[1] The high computing complexity from MIMO functions, however,

[1]MIMO technology enables the feasibility of high-speed wireless connectivity (0.1 to 1.0 gigabit per second (Gbps)) by employing multiple transmit and receive antennas to simultaneously access multiple spatial streams, achieving high spectral efficiency of around 50bps/Hz (Fig. 1.2). Alternatively, MIMO systems can exploit the diversity gain by sending multiple copies (or slight modifications) of the same data stream through several transmit antennas, resulting in independently-faded replicas from the same signal source. The receiver then constructively combines these signal replicas to gain reliable transmission.

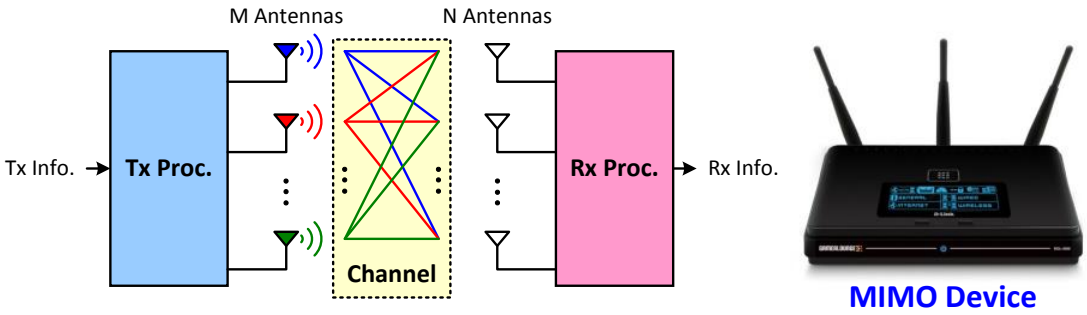


Figure 1.2: MIMO communication systems.

incurs very high power consumption (low energy efficiency) and a high silicon-area cost, shortening the battery life and increasing the fabrication cost. Maintaining high efficiency and low power density is therefore of great importance for mobile terminals such as cellular phones.

To summarize, the diverse standardization and the proliferation of portable electronics in radio applications have driven the need for *flexible* and *energy-efficient* signal processing. The number of standards a single device can dynamically support with low energy budget is becoming the key to product success.

1.2 Software-Defined Radios

The requirements for flexibility and efficiency lead to the concept of Software-Defined Radios (SDRs), where the components that used to be hard-wired (e.g. channel estimation, filtering, signal equalization) are replaced by flexible hardware to dynamically adapt to evolving protocols (Fig. 1.3). With the flexibility provided by SDRs,

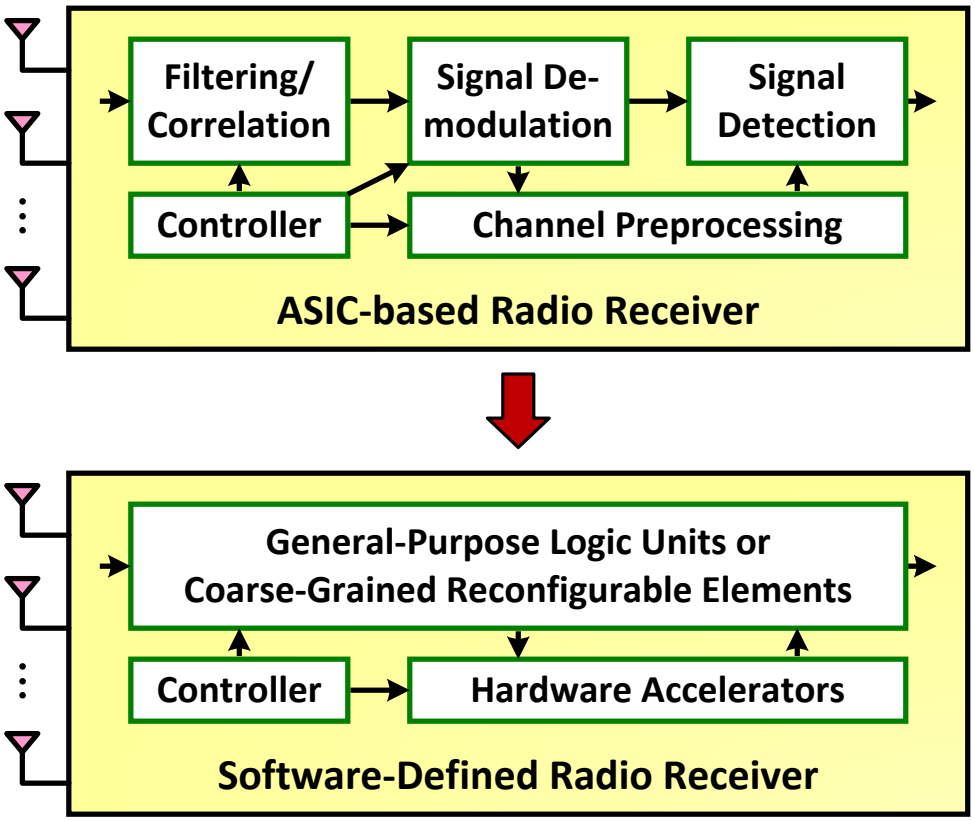


Figure 1.3: Software-defined radios replace the dedicated modules for fixed applications/standards to more flexible hardware to adapt to volatile channel environments and standards. Sometimes a dedicated accelerator is still required to speed up certain computationally demanding tasks.

a variety of communication bands can be used based on the software or configuration patterns.

Existing SDRs exploit (1) various levels of parallelism, (2) coarse-grained kernels, and (3) multi-core architectures to maintain high energy efficiency. From architecture perspective, these designs can be further categorized into two types [1]: (1) Processor-based architectures [2]-[8] (Fig. 1.4), and (2) Coarse-Grained Reconfigurable Architec-

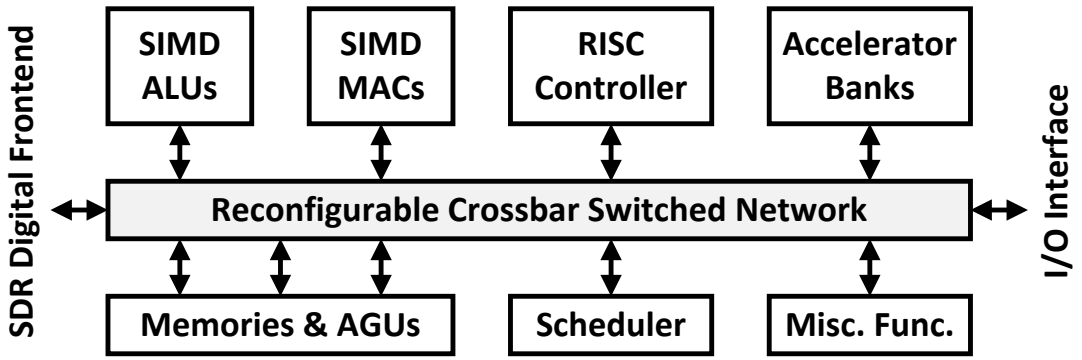


Figure 1.4: Processor-based SDRs exploits DLP/ILP/TLP, versatile memory access, and multi-core communications for high energy efficiency.

tures (CGRAs) [9]-[12] (Fig. 1.5).

1.2.1 Processor-based SDR

The processor-based SDRs pay more attention on the (Data-/Instruction-/Task-Level) Parallelism (DLP/ILP/TLP) and multi-core communications. It typically consists of multiple real- or complex-valued Multiplication-ACcumulation (MAC) blocks and Arithmetic-Logic Units (ALUs), some memory banks, an on-chip network, several accelerators, and a central controller unit. A single software instruction can drive multiple data and/or tasks (namely single-instruction-multiple-data/-task (SIMD/SIMT), so the energy overhead from the instruction memory can be diluted. SIMD is also very suitable for vector and matrix operations that are essential to most of the radio algorithms. The controller efficiently manages the SIMD MACs/ALUs and the memories so that multiple threads can be launched simultaneously. Each memory also contains its own Address-Generation Unit (AGU) to minimize memory access conflicts and to enable

Direct-Memory Access (DMA). The programmable reconfigurable crossbar switches are used as the on-chip network for core-to-core communications. Two of the processor designs deserve more attention: (1) The MPSoC Tomahawk [8] exploits not only SIMD/SIMT but also has a dedicated run-time scheduler hardware unit (CoreManager) for dynamic power and workload management; (2) The ConnX BBE by Tensilica Inc., different from other fixed SDR products, is offered as a modifiable hardware IP. The users are allowed to modify the functional blocks and the control mechanism at design time by using Tensilica Xtensa template processor as a foundation. Different processor configurations according to the application requirements are generated using tools like Xtensa Processor Generator and Tensilica Instruction Extension. Such freedom helps shorten the time-to-market yet preserve quite some degrees of creativity and flexibility for the designers.

1.2.2 CGRA-based SDR

As opposed to the processor-based SDRs, the CGRAs focus more on the core granularity to optimize the data locality (i.e. reducing the frequent data movement between Compute Elements (CEs) and program memories as typically seen in processors). Note that CGRAs inherently imply multi-core structure and task-level parallelism. The Architecture for Dynamically Reconfigurable Embedded Systems (ADRES) from IMEC. is a classic example of the CGRA. It tightly couples a Very-Long-Instruction-Word (VLIW) processor and a coarse-grained reconfigurable matrix. This tightly-coupled system has the advantages of shared resources, reduced communication costs, improved

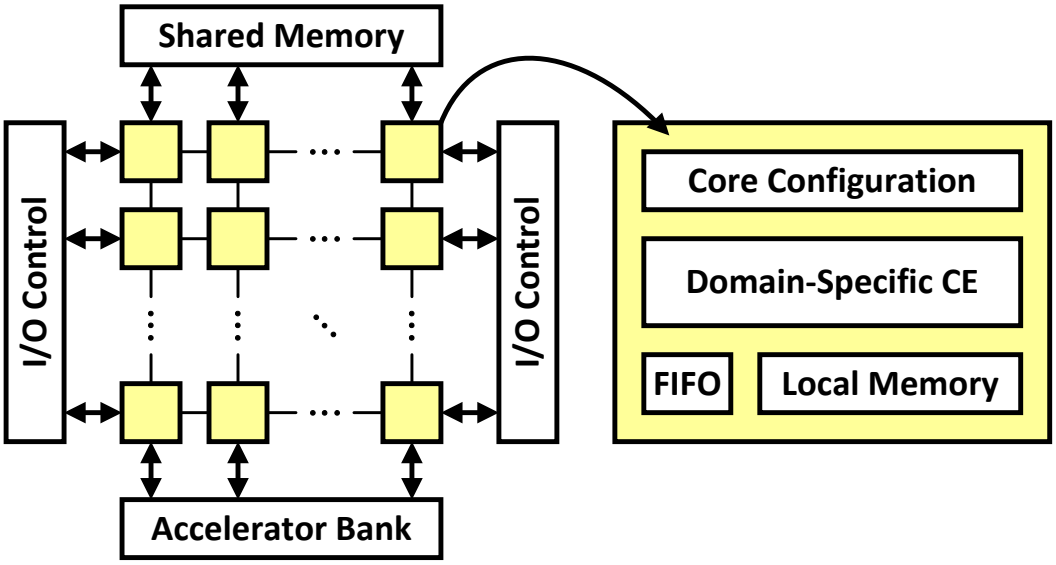


Figure 1.5: CGRA-based SDRs exploits DLP/TLP, multi-core communications and domain-specific, coarse-grained compute elements for high energy efficiency.

performance, and simplified programming model. The VLIW processor and the reconfigurable matrix share CEs and local Register Files (RFs). For the reconfigurable matrix part, there are many Reconfigurable Cells (RCs) which comprise CEs, RFs, and the configuration memory. The RCs are connected to the nearest neighbor RCs and RCs within the same row or column in the tile. Therefore, kernels with a high level of DLP are assigned to the computing elements, whereas sequential codes are run on the VLIW processor as central control. The data communication is performed through the shared RF, which is more compiler-friendly than the message-passing method as seen in the processor-based SDRs.

1.3 Cognitive Radios

The other hot topic for future radio implementation is the cognitive radio (CR). The concept of CR is very similar to SDR, in that they all pursuit flexible hardware adjustments for optimal system performance. The CR, however, considers not only the hardware efficiency but the *spectral* efficiency. Specifically, since the wireless standards typically have to operate on a fixed spectrum assignment, the spectrum is severely under-utilized. A research team measured the frequency bands <3GHz from Jan. 2004 to Aug. 2005 and concluded that the average spectrum occupancy is only around 5.2% [13]. Another research from Berkeley Wireless Research Center (BWRC) showed the frequency band beyond 2GHz is highly under-utilized (Fig. 1.6) [14]. The CR, as a result, is defined as a system which is (1) aware of its surrounding electromagnetic environment and (2) intelligent to improve the spectral efficiency by dynamically sensing and utilizing the unused spectrum holes [15]. The CR also has to be aware of the presence of legitimate users with higher priorities of spectrum usage, so that the secondary users won't create harmful interference.

1.3.1 Spectrum Sensing and Blind Signal Classification

Spectrum sensing and signal classification are the most important steps for CRs. Under predefined constraints of detection probability and processing time, the spectrum sensing helps to detect the presence of primary users within the Band of Interest (BOI). The signal classification, on the other hand, has numerous applications in current

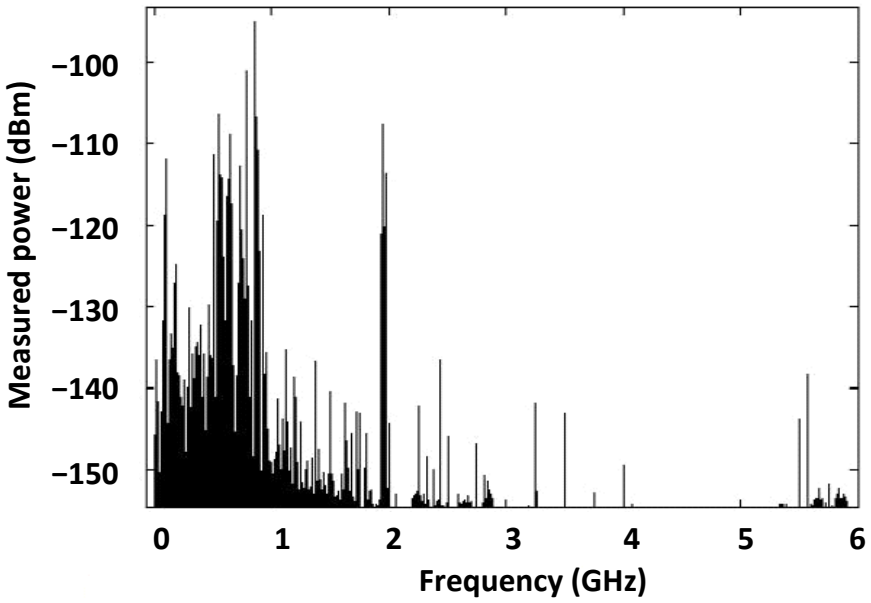


Figure 1.6: Measurement of 0-6 GHz spectrum utilization at BWRC [14].

and future wireless networks. From an electronic surveillance point of view, military applications of blind modulation classifier include tracking the spectrum activity of specific users (often interferes or jammers) and learning their modulation classes. Signal classification is therefore vital to electronic countermeasures in such hostile environments. Additionally, with the recent deployment of heterogeneous networks (HetNet) such as long term evolution (LTE), modulation classification becomes part of interference management [18], [19]. Multi-user detection is performed to support multiple overlapping transmissions in time and/or frequency. Knowledge of the modulation type by means of modulation classification [20] is necessary to demodulate the interfering signal [21]. However, as a result of dynamic signal allocation on any spectrum holes from the CR technology [22], information about the transmit parameters and the modulation schemes at any frequency band can no longer be assumed (as compared to the

primal cases where the wireless standards operate on a fixed spectrum assignment). In such future wireless applications, ***blind signal classification*** is of significant research interest.

1.4 Motivation of This Work

Several remaining challenges from SDRs and CRs motivate this research.

1.4.1 Tradeoff Between Efficiency and Flexibility

Albeit claimed as highly-efficient, existing SDR architectures are in fact *much less* efficient than dedicated hardware. A survey in [23] pointed out the efficiency of SDRs is no more than 0.1 billion-operation-per-second per milliwatt (GOPS/mW) (Fig. 1.7), which is 40 to 150$\times$ lower than the typical ASIC efficiency of 4 to 15GOPS/mW at nominal voltage in deep-submicron (e.g. 65nm) technology. The primary reason is the inherent trade-off between the flexibility and efficiency.

Flexibility and efficiency are conflicting criteria. In general, the energy and the area efficiency are measured as GOPS/mW and GOPS/mm^2, where only the arithmetic blocks (i.e. multiplication and addition) are considered as effective operations[2]. Flexibility is hard to fairly quantify, but in a broad sense it comes with added datapath and control complexity to the original, less-flexible circuit. Since the control circuits (e.g. multiplexers, state machines, instruction memories, etc.) incur power/area overhead

[2]We consider one 16-bit (16b) real-valued addition (ADD) or its equivalents as one operation. For example, a 16b$\times$16b array-based multiplication is equivalent to 15 operations.

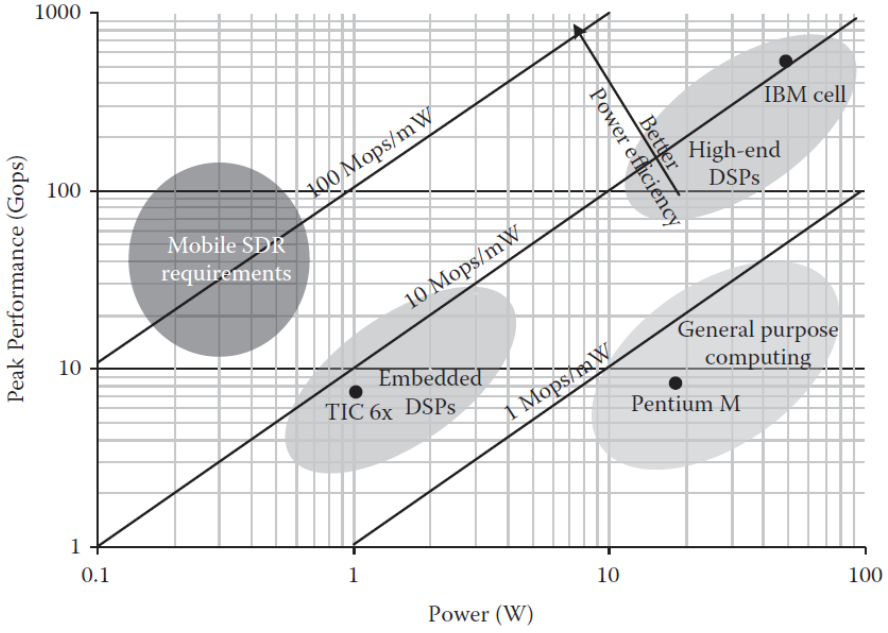


Figure 1.7: Throughput and power requirements of typical 3G wireless protocols. The results are calculated for 16b fixed-point operations [23].

but are not considered as effective operations, increasing a circuit's flexibility implies the loss of its efficiency. Figure 1.8 illustrates the idea by showing the generic trend between various types of implementations [24]. On average, the dedicated hardware (i.e. application-specific integrated circuit, ASIC) exhibits the highest efficiency, while the microprocessors are advantageous in their flexibility. Programmable digital signal processors (DSPs) are a viable compromise but still exhibit a large efficiency gap ($>10\times$) to ASICs. In this dissertation, we propose a new design that closely matches the efficiency of ASICs while keeping the flexibility of DSPs. Techniques to achieve this goal are presented through several design examples.

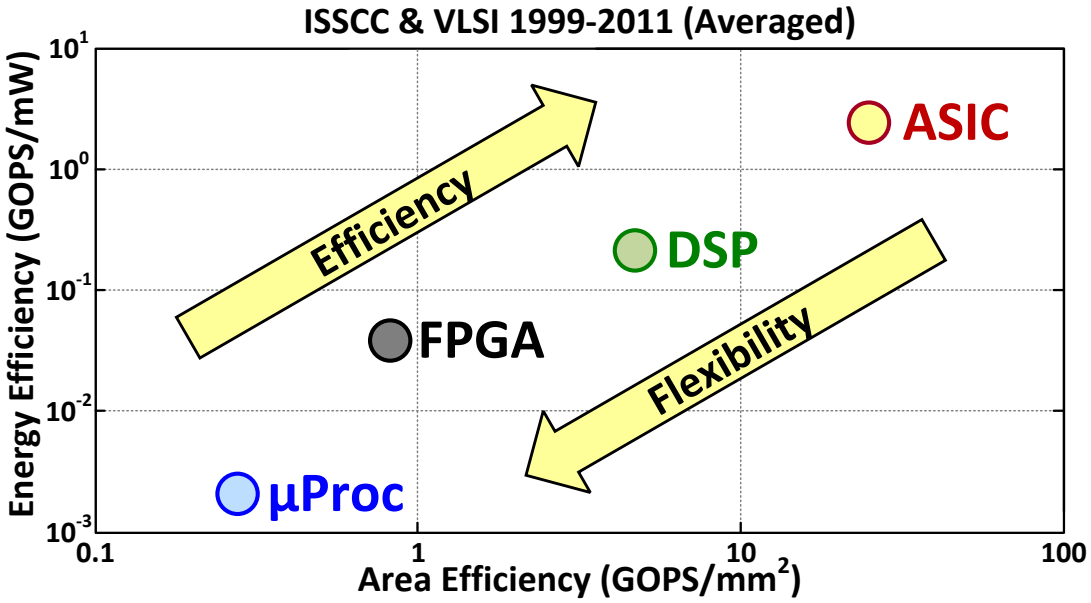


Figure 1.8: Tradeoff between efficiency and flexibility across various types of implementations [24].

1.4.2 Wideband Spectrum Sensing and Blind Signal Classification

Wideband (>100MHz) sensing is a highly desirable feature of CRs, since it allows simultaneous sensing over multiple BOIs and thereby increasing the detection probability and system throughput. Wideband sensing, however, imposes many design challenges to the digital baseband processing [25]-[31]. In the digital baseband, the sensing hardware needs to provide reliable signal detection in a negative SNR regime while operating in real time. The DSP baseband, as a result, must accommodate advanced signal processing algorithms within limited power and area while still meeting the performance constraints such as detection probability and sensing time.

The key challenge of blind classification, on the other hand, is how to minimize

the energy given the absence of *a priori* information about the transmit parameters, i.e. the carrier frequency (F_c) and the symbol rate (F_s). Most commonly used classifiers are based on the detection of cyclostationary features, which are second-order moments of a signal, related to its F_c and F_s. Existing classification frameworks assume standard-compliant signals with known parameters, so the modulation classification can be performed through exhaustive search of known signal features [18]. However, as mentioned in the previous section, the signal parameters in a real CR system is unknown, making traditionally exhaustive search methods energy inefficient due to high computational complexity. Such exhaustive search of features in a wideband channel is even more energy inefficient and unsuitable for real-time processing.

From an architectural perspective, although various non-blind classification algorithms have been studied and even implemented on DSP [40] and SDR platforms [41], an efficient silicon realization that classifies multi-carrier, spread-spectrum, and linearly-modulated signals was never realized before. Again, these classifiers require *a priori* knowledge of the signals of interest, making them unsuitable for real-time blind classifiers. In order to achieve high energy efficiency, ASIC implementation is desirable. However, due to diversity of modulation classes and algorithms for the classification, a heuristic ASIC design equipped with multiple dedicated modules – one for each signal class – would result in a large area cost due to the difficulty of hardware sharing.

Another issue is the dynamic range of the possible signal bandwidth. In an ASIC design, the optimal voltage, clock rate and architectural parallelism is a function of the

target system throughput. However, as the signal bandwidth is unknown in a classification system, the throughput is volatile. A classification system, as a result, should dynamically match its voltage, clock rate and parallelism to the detected signal bandwidth to optimize its energy efficiency, posing a challenge to chip design.

This dissertation presents the world's first wideband, real-time blind classification processor for CRs. We adopt an FFT-based band segmentation engine with associated partial-spectrum sensing schemes for reliable parameter estimation and energy-efficient implementation. We also propose a reconfigurable feature extraction engine with high functional diversity and energy/area efficiency. By jointly considering the algorithm and architecture layers, we first select computationally efficient parameter estimation and modulation classification algorithms. We then exploit the functional similarities between algorithms to build a processing architecture that maximizes hardware utilization. We carefully analyze the processing strategy and the programming model of the processor to minimize the overall energy. Circuit-level techniques such as Dynamic Parallelism and Frequency Scaling (DPFS), high-performance power gating, and multi-core dynamic scheduling are also applied to always achieve the optimal energy point regardless of the signal bandwidth.

1.5 Dissertation Outline

This work adopts several techniques to the proposed software-defined and cognitive radio system, with the goal to achieve near-ASIC energy efficiency without giving up

the flexibility of a DSP. The techniques are (1) Algorithm-architecture co-design that includes workload analysis to set the proper design constraints to drive the decision on domain-specific dataflow structures; (2) Multi-core dynamic scheduling that includes dynamic parallelism-frequency scaling (DPFS) and aggressive power gating to achieve the optimal energy efficiency which is seemingly invariant to throughput changes; (3) Flexible instruction-set architecture (ISA) and programmable state machine (SM) to simplify the traditional ISA structure in the processors, lower the program runtime and improve the adaptability to design changes; (4) Multi-scale on-chip interconnects that includes local uni-directional fastpath and global hierarchical network for high throughput data arbitration and I/O interfacing. Chapter 2 explains the efficiency and flexibility in more detail, and gives more insights about the aforementioned techniques that break the efficiency-flexibility tradeoff. As a proof of concept, Chapter 3 describes the design of a wideband blind signal classification processor for CRs. Another example, presented in Chapter 4, is a 16-core processor for baseband SDR signal processing. The blind classification processor achieves a 59$\times$ energy saving compared to an exhaustive method, while the 16-core processor shows $>$2.4$\times$ higher energy efficiency than state-of-the-art communication processors and closes the gap with functionally-equivalent ASICs to within 2.6$\times$. Chapter 5 highlights the chip verification process. Last but not least, the techniques proposed in the dissertation not only enable energy-efficient and flexible radio implementation, but can also be applied to other domains of computing. As a result, Chapter 6 concludes this work and discusses future research directions.

CHAPTER 2

Efficiency and Flexibility

This chapter presents the efficiency and flexibility in more details, and highlights the ways to quantify these two criteria. It also gives some insights about why there exists inherent tradeoff between the efficiency and flexibility. Lastly, some possible solutions to break the tradeoff are highlighted.

2.1 Definitions and Limits

Future VLSI designs require both the flexibility and the efficiency to achieve truly multi-mode and energy/area-efficient MIMO signal processing. Following equations and definitions are presented to quantify the efficiency:

$$\text{AreaEfficiency} = \text{GOPS}/\text{mm}^2, \tag{2.1}$$

$$\text{EnergyEfficiency} = \text{GOPS}/\text{mW}, \tag{2.2}$$

where GOPS stands for "billion operations per second", and each operation is defined as a 16-bit (16b) addition operation or its equivalents. For example, a 16b$\times$16b array multiplication is considered as 15 operations. Higher efficiency means better control of power budget or more economic usage of silicon area. Since only the addition or

its equivalents are considered as effective operations when we calculate the efficiency, there naturally exists a technology-dependent upper bound of energy efficiency. In the 40nm regular-V_{th} CMOS process, for example, the raw efficiency of a 16b adder is around 25GOPS/mW, and it goes up to 50-55GOPS/mW when operating at near-threshold supply voltage. Such theoretical upper bound can serve as the reference point for us to quantify the energy overhead of various types of designs. For instance, if a design has about 5-10GOPS/mW at nominal voltage in 40nm process, it can be fairly quantified as having an energy overhead of 2.5-5$\times$ from the control and datapath circuits.

The flexibility is generally quantified by four degrees of freedom, i.e. parameter, function, algorithm, and standard. As shown in Fig. 2.1, a wireless standard has its own set of parameters and functions, and each function can be realized by multiple (usually more than one) algorithmic candidates. The standard-scope flexibility shows how many standards a device can support, while the flexibility in the other three degrees of freedom measures the ability of dynamic resource allocation within a standard. By dynamically redistributing available resources, the focus can either be on mobility management or high data rate. Taking the signal detection part in Fig. 2.1 as an example, it is shown that the function can be realized by the zero-forcing (ZF), the minimum-mean-square-error (MMSE), or the sphere detecting (SD) algorithm. Under the severely-fading environment, the hardware can run advanced (yet computationally intensive) algorithms (i.e. SD) to keep the quality of communication. When the channel condition becomes better, simple algorithms (i.e. ZF) are con- sidered to increase the throughput of the

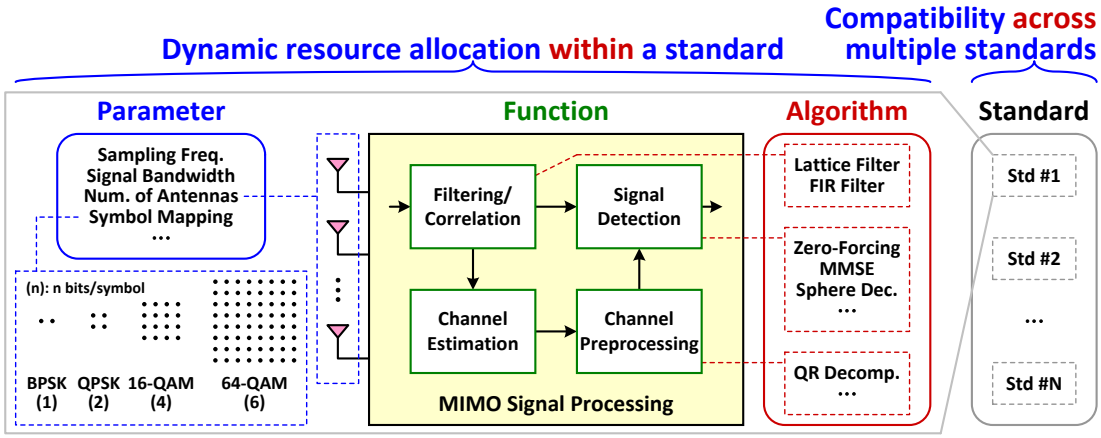


Figure 2.1: Four degrees of flexibility: parameter, function, algorithm, and standard.

system. Since the channel condition changes all the time and is unpredictable, it is believed that better performance can be achieved by applying flexibility. In addition, the high development cost ($50M in 40nm CMOS process) and long time-to-market (>1 year) associated with dedicated hardware designs can be avoided [16].

2.2 Inherent Tradeoff

Many existing hardware solutions try to gain both flexibility and efficiency, but none of them has thus far reached the goal. These solutions can be categorized into two groups, namely the programmable digital-signal processors (DSPs) and the dedicated application-specific integrated circuits (ASICs). Figure 2.2 shows the efficiency plot of selected chips for multi-antenna signal processing. All of the designs in the plot are normalized to a 65nm CMOS technology for fair comparison. It is clear that ASICs have high efficiency but only explore the parameter-scope and some parts of the standard-scope flexibility. The DSPs, on the other hand, are highly flexible but have

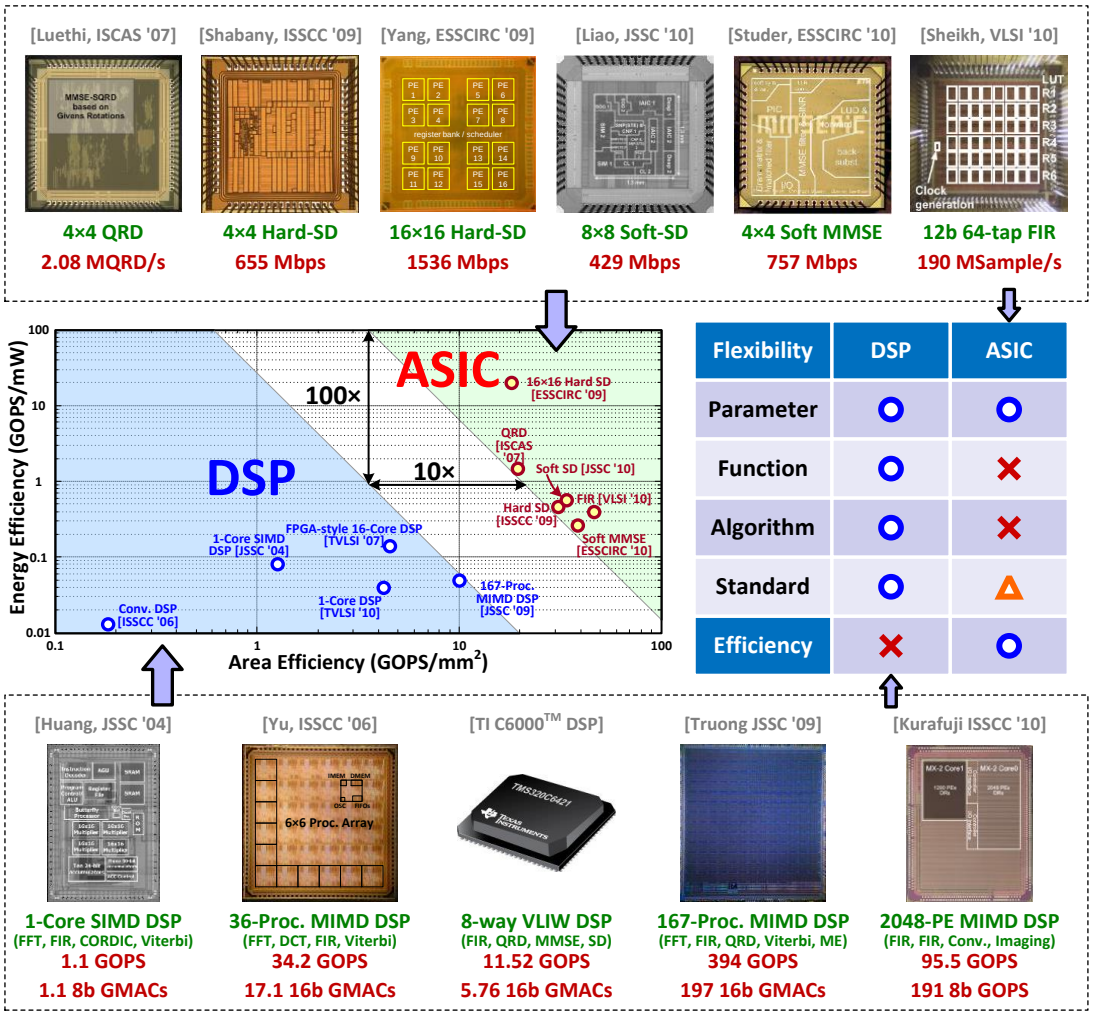


Figure 2.2: Existing chip designs for MIMO signal processing.

low efficiency compared to ASICs. The other concern for DSPs is about the efficiency gap. As highlighted in Fig. 2.2, there exists at least a 100× and a 10× gap in energy and area, respectively. The performance of 65nm DSPs in such situation only equals the ASICs in 180nm or older technologies, at least three generations behind (from 180 to 65 nm). Also the fabrication cost in 65nm process is around 7× of that in 180nm process [17], meaning that a company has to pay 7× more to buy flexibility with a performance of an old-generation dedicated chip.

2.3 Techniques for High Efficiency and Flexibility

The aforementioned problems motivate this research, which aims at investigating a new VLSI architecture that matches efficiency/throughput of ASICs but keeps flexibility of DSPs. The method to achieve this goal is to *hybridize* the design concepts for DSPs and ASICs.

The instruction memory, the control circuits, the low utilization of processing kernels, and the mismatch between algorithm-architecture spatial mapping are the four major factors that cause the efficiency loss. Dedicated hardware generally doesn't need an instruction memory and data memory for data movement. Instead, its datapath is carefully designed to perform "in-place" processing. Consequently, ASICs are more power- and area-efficient, and potentially achieve lower processing time because they don't need to spend time on confirming the data availability. The low utilization of processing kernels comes from the fact of variable workloads in programmable processors. Sometimes the processors are doing compute-intensive jobs, at other times most of theirs cores are in idle states. If the hardware doesn't have power-management feature to opportunistically switch off the under-utilized kernels, the leakage current will eventually contribute significant energy overhead to the chip. This phenomenon is even more severe in today's deep-submicron CMOS technology. The inefficient datapath design due to the mismatch between algorithm-architecture mapping incurs energy overhead as well. The general-purpose RISC processor is a extreme case, whose datapath tries to perform complex functions by combining multiple simple and elementary

instructions. Although this concept is effective to cover all of the possible functions, it creates dramatically large energy overhead from the instruction memory that drives the datapath. The RISC datapath also requires a lot more processing cycles compared to the ASICs, so in general the processor needs to operate at a much higher clock frequency to compensate the loss in throughput, further increasing the switching power.

In the rest of the dissertation, we propose the following techniques to enable the co-existence of flexibility and efficiency in one chip. Simply speaking, the new architecture will consist of ASIC-like processing kernels plus efficient control strategy to achieve flexible processing.

Algorithm-architecture co-design: The algorithm-architecture co-design involves careful selection of robust, hardware-friendly, and architecturally similar algorithms with the goal to enable high degree of hardware reuse and determine the proper granularity. Sometimes the algorithms might be dependent with each others, so we also need to find the proper programming model (i.e. processing strategy) to efficiently control the hardware. This technique is used for both the chip designs for the blind signal classification and the SDR baseband DSP.

Dynamic multi-core management and parallelism-frequency scaling: Traditional ASIC design optimization is based on a pre-defined throughput. Specifically, the designer can easily decide optimal combination of the clock frequency, the supply voltage, and the degrees of parallelism to achieve the highest energy efficiency that meets the target throughput. However, in a flexible implementation that focuses on *variable* throughput, the above optimal combination will no longer be the best solution

for all cases. Instead, a more advanced scheduling algorithms and the dynamic parallelism, voltage, and frequency scaling have to be jointly applied to keep the circuit's efficiency always high regardless of the throughput requirements. To achieve this goal, we will do aggressive power gating to turn off unused processing elements, and/or try to make all of the processing elements busy to handle independent data threads. The dynamic multi-core management strategy will be demonstrated in the next chapter for the multi-signal blind classification.

Flexible instruction-set architecture: Reducing the control overhead is the most effective way to approach the ASIC's performance and efficiency. Instead of using the traditional instruction set architecture that is complex and has to be defined at design time, we propose a flexible ISA that enables the freedom to define the proper control patterns at system run-time. This concept starts with the observation that, for each particular task within the supported application domain, only a subset of the entire instruction set is required. If we can flexibly define *only* the necessary instructions prior to executing a task, then we no longer need a fixed and complex ISA to support all possible control patterns. Adaptation problems with design changes can also be resolved by simple hardware reconfigurations. We will demonstrate this concept in the 16-core SDR baseband DSP, where the nature of SDR workloads is exploited for energy-efficient and flexible kernel control. A flexible ISA that can adjust itself to satisfy the task-specific needs will be shown as more efficient than a complex hard-wired ISA with high instruction coverage but low utilization.

Multi-scale on-chip interconnects: Efficient interconnects are essential for robust

and high-throughput data processing for the multi-core processor implementations. In the 16-core SDR DSP, we propose a multi-scale interconnects with uni-directional local fastpath and the radix-2 hierarchical global network to allow global data exchange and multicasting between cores and external interfaces.

2.4 Summary

Energy efficienty and flexiblity are conflicting design requirements. We outline four promising solutions to achieve both the flexibility and efficiency in one chip. They are:

- Algorithm-architecture co-design
- Dynamic multi-core management and parallelism-frequency scaling
- Flexible instruction-set architecture
- Multi-scale interconnects

These techniques will be demonstrated in the following chapters, on two chip design examples.

CHAPTER 3

Design Example 1: A 500MHz Wideband Blind Classification Processor

This chapter demonstrates an energy-efficient, wideband blind classification processor. We focus on a channel bandwidth of 500MHz, and consider a minimum signal-to-noise ratio (SNR) of $\geq$0dB. This range of SNR is reasonable for classification of interferers in multi-user detection and blind signal demodulation applications. Detailed design specifications are summarized in Table 3.1. The frequency resolution is set to 12.5kHz to detect narrowband interferers. The classifier should identify (1) multi-carrier (MC) orthogonal-frequency-division-multiplexing (OFDM), (2) direct-sequence spread spectrum (DSSS), and (3) linearly-modulated signal-carrier QAM/PSK/MSK signals with a detection probability of $\geq$95% and a false alarm rate of $\leq$0.5%. The processor needs to meet an energy constraint of 400μJ and a processing time of 20ms at 0dB, and 20μJ and 20ms at 10dB SNR.

As mentioned in Chapter 1, the key challenge of blind classification is how to minimize the energy given the absence of *a priori* information about the transmit parameters, i.e. the carrier frequency (F_c) and the symbol rate (F_s). Whatever algorithms are

Table 3.1: Design specifications of the blind classification processor.

Variables	Specifications
Modulation Classes	OFDM, DSSS, M-QAM, M-PSK, MSK
Probability of Correct Classification	$\geq 95\%$
Probability of False Alarm	$\leq 0.5\%$
Channel Bandwidth	500MHz
Signal Bandwidth	$\leq$ 125MHz
Frequency Resolution	12.5KHz
Minimum SNR	$\geq$ 0dB
Energy Budget	400μJ (0dB); 20μJ (10dB)
Processing Time Budget	20ms (0dB), 2ms (10dB)

used, an exhaustive search of signals' features in a 500MHz wideband channel with uncertain F_c/F_s is energy inefficient and unsuitable for real-time processing. As a result, the proposed blind classification processor features a three-step (coarse-fine-residual) parameter estimation for a 59$\times$ energy saving compared to an exhaustive method. An FFT-based band segmentation (BSG) engine performs the coarse and fine parameter estimation, followed by the proper down-conversion and down-sampling by the digital mixers and the filters, respectively [43]. The reconstructed signal is then sent to the feature extraction (FEX) engine for residual parameter estimation and signal classification [46]. Figure 3.1 illustrates the processor architecture, system flow, and example waveforms of parameter estimation.

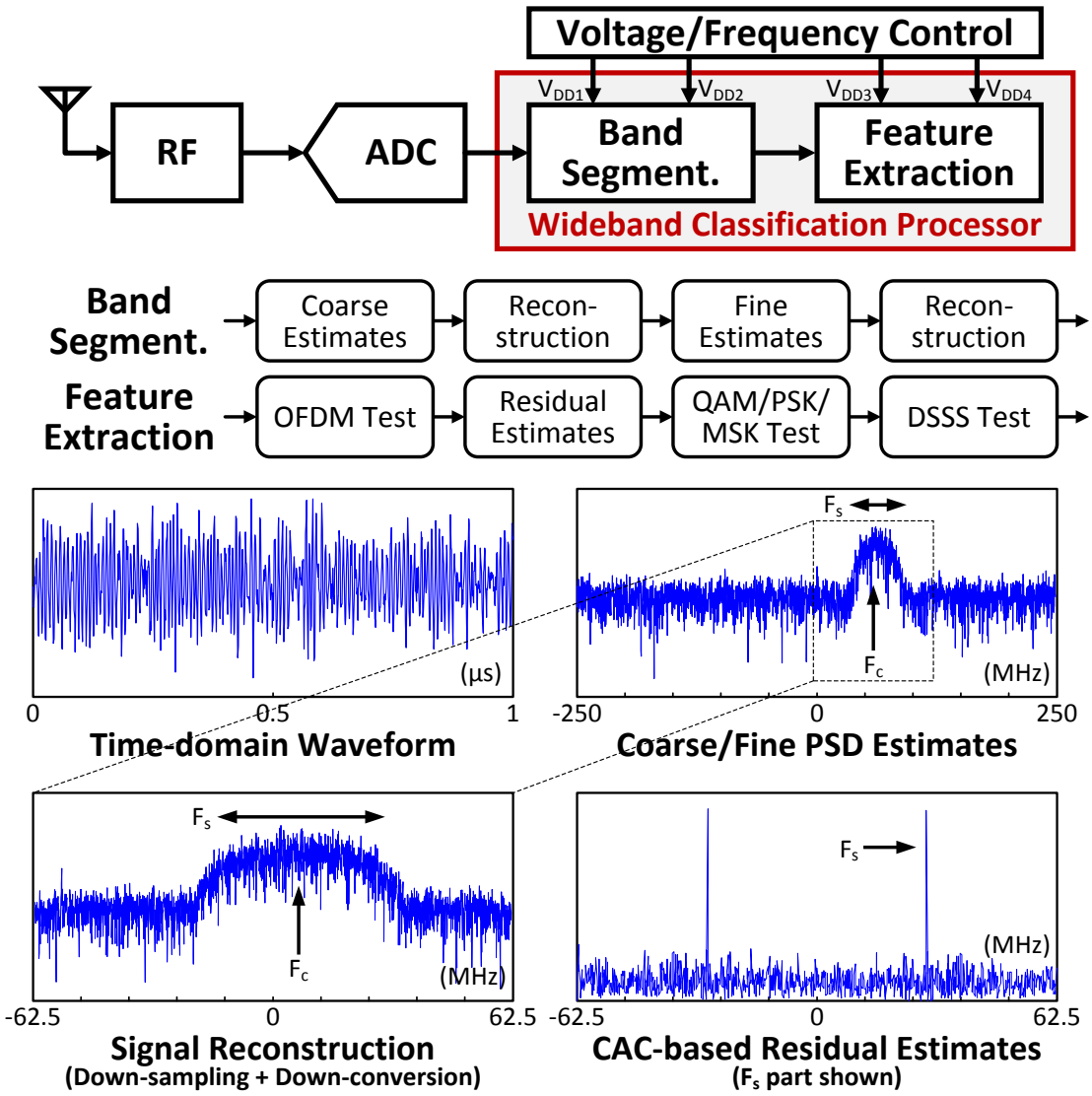
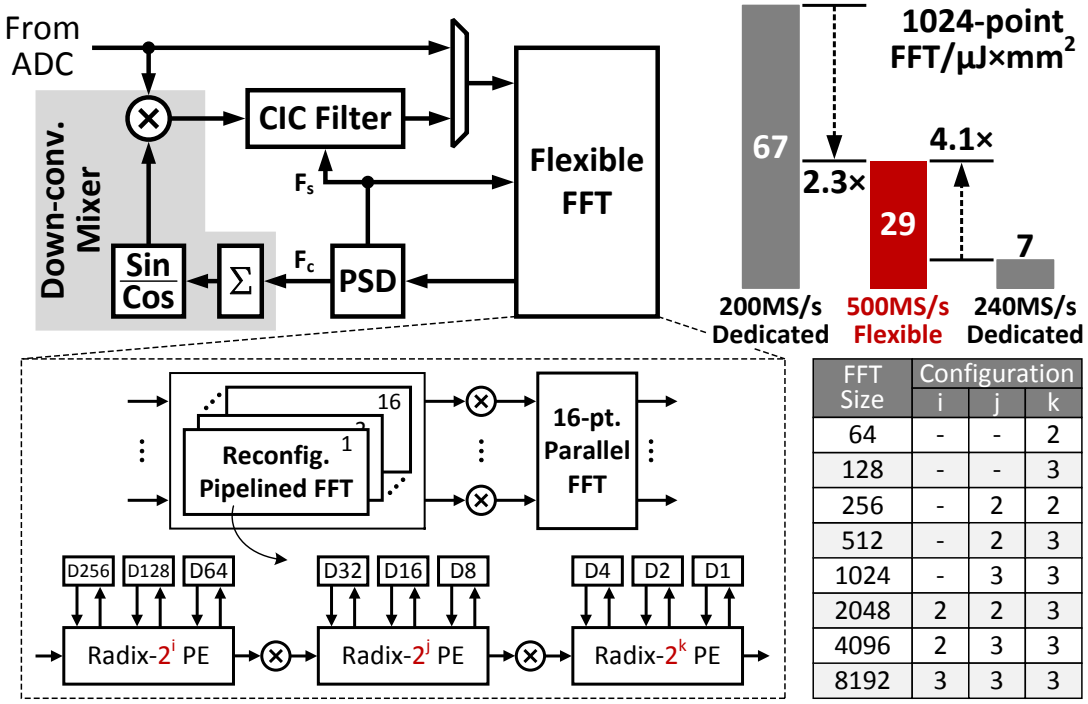


Figure 3.1: Architecture, system flow, and example waveforms of parameter estimation (F_c=60MHz, F_s=30MHz, Channel=500MHz) of the classification processor. The cyclic-autocorrelation (CAC) function is adopted for residual parameter estimation.

In the remainder of the chapter, we first review the BSG engine that handles the coarse and fine parameter estimation. After that, we describe the low-complexity classification algorithms, and an algorithm-architecture co-design methodology that considers the tradeoffs among *dependent* blocks of the FEX engine. A reconfigurable architecture for signal classification is then proposed based on the algorithm-architecture co-design framework and the energy-efficient circuit techniques. Two chip implementations are presented (the second is an improved version of the first). We show the potential benefits of dynamic parallelism and frequency scheduling from the measurement results of the first chip, and present a second version that implements multi-core dynamic scheduling and power gating for multi-signal classification under a multi-path channel.

3.1 Band Segmentation Engine

The BSG, consuming $>$70% of the total energy, is the key to energy-efficient blind classification [43]. It shrinks the exhaustive F_c/F_s search range of 500MHz to 62.5kHz, an 8000$\times$ reduction, by employing a flexible FFT and a power-spectral-density (PSD) detector (Fig. 3.2). The signal reconstruction is supported by a down-conversion mixer and a flexible cascaded integrator-comb (CIC) filter [45]. Heuristically, using an 8192-point FFT to analyze the full channel for once (full-PSD sensing) can deliver reliable F_c/F_s estimates for the FEX. We propose a coarse-fine (partial-PSD sensing) scheme to further reduce the BSG energy. The idea is to split the 8192-point FFT into smaller



FFT Size	Configuration		
	i	j	k
64	-	-	2
128	-	-	3
256	-	2	2
512	-	2	3
1024	-	3	3
2048	2	2	3
4096	2	3	3
8192	3	3	3

Figure 3.2: Band segmentation engine with partial-PSD sensing.

FFTs for equivalent performance, since a typical narrow-band (≈10MHz) unknown signal induces too much energy waste on the null frequency bins in the full-PSD scheme (due to the nature of spectrum under-utilization).

A systematic tradeoff analysis is performed to determine the energy-optimal combination of the coarse- and fine-sensing FFT sizes for a given F_s [43]. We conclude that a 64-point coarse plus a variable-length (512-8192) fine FFT yields up to 3.4× energy saving when 4MHz<F_s<65MHz (Fig. 3.3). For F_s outside this range, the BSG energy overhead is only 0.4% compared with the full-PSD approach. As the F_s of CR signals typically falls in the range between 5 and 50MHz, the partial PSD is statistically more energy efficient, yet it preserves the flexibility to handle wide ranges of F_s with

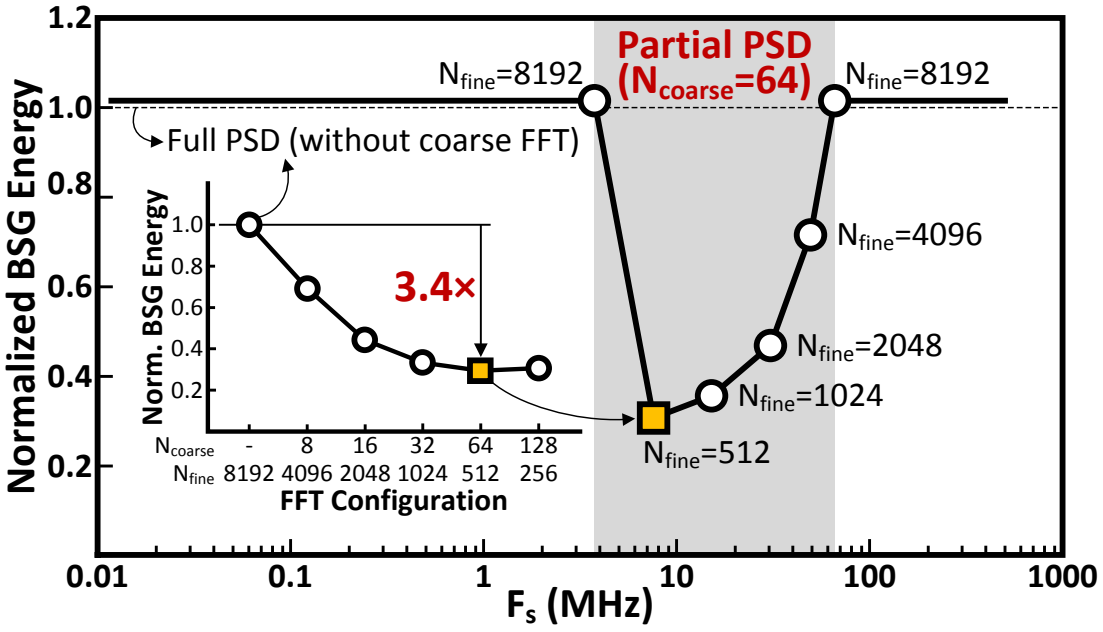


Figure 3.3: Band segmentation engine with partial-PSD sensing.

negligible overhead.

A 64- to 8192-point reconfigurable FFT, reused for both the coarse and fine parameter estimates, is implemented by 16 banks of 4- to 512-point pipelined FFTs followed by a 16-point parallel FFT for a minimum energy-area product (EAP) (Fig. 3.2). This multipath mix-radix flexible architecture shows a 4.1× lower EAP than a dedicated 1024-point radix-4 FFT that minimizes the subthreshold energy [47] . The flexibility, however, causes a 2.3× EAP overhead compared with our previous dedicated 1024-point FFT that uses the same design methodology [48] .

3.2 Classification Algorithms

In this section, we present the proposed algorithmic hierarchical classification tree. The design hierarchy is based on both the level of *a priori* information that a block requires and its computational complexity. In particular, the blocks that do not require *a priori* information about the signal being classified are processed first. For instance, the OFDM classifier employs a totally blind low-complexity algorithm, and therefore can be performed first. This design methodology dictates the order in which the classification algorithms are performed as shown in Fig. 3.1.

3.2.1 Multicarrier Classification

This block differentiates between multicarrier (MC) and single carrier (SC) signals. The MC classifier is based on the fourth-order cumulant C_{42} [49] which is a form of a Gaussianity test. The property of C_{42} is that it converges to zero if the input samples are approaching Gaussian distribution. The C_{42} statistic of an OFDM signal, as a result, is close to **zero** since the OFDM is a mixture of a large number of sub-carrier waveforms. For other narrowband SC signals, the test statistic converges to a **non-zero** value, thereby making it possible to separate MC from SC signals without any information about the signal's carrier frequency and symbol rate. The fourth-order cumulant is computed as follows:

$$C_{42} = \frac{1}{N_m} \sum_{n=1}^{N_m} |x[n]|^4 - |C_{20}|^2 - 2C_{21}^2, \tag{3.1}$$

where N_m are the number of samples used for distinguishing MC and SC signals, $x[\cdot]$ is the vector of samples obtained from the CIC filter, $C_{20} = \frac{1}{N_m}\sum_{n=1}^{N_m} x[n]^2$ and $C_{21} = \frac{1}{N_m}\sum_{n=1}^{N_m} |x[n]|^2$. The MC detection is a threshold-based test derived by comparing C_{42} to an SNR-dependent threshold γ_m. The algorithm has low computational complexity and classification time since only $N_m = 100$ samples are required to guarantee a classification probability of 95% for MC signals at 10dB SNR.

3.2.2 Residual Carrier Frequency and Symbol Rate Estimation

The residual F_c/F_s estimation is necessary before the modulation-type classification of SC signals. This is because the fine estimation from BSG only guarantees a resolution down to 62.5kHz, resulting in an estimation error of $\pm 2.5\times$ of the minimum system resolution (=12.5kHz). Such error will greatly degrade the performance of the modulation-type classification [50].

Both the residual estimation and the modulation-type classification for SC signals rely on the cyclic autocorrelation (CAC) function to detect their cyclostationary features. Under a finite number of samples N, the conjugate and the non-conjugate CACs can be computed respectively as follows:

$$\tilde{R}^{\alpha}_{x^*}(\nu) = \frac{1}{N}\sum_{n=0}^{N-1} x[n]x^*[n-\nu]e^{-j2\pi\alpha n T_s}, \tag{3.2}$$

$$\tilde{R}^{\alpha}_{x}(\nu) = \frac{1}{N}\sum_{n=0}^{N-1} x[n]x[n-\nu]e^{-j2\pi\alpha n T_s}, \tag{3.3}$$

where ν is the lag variable, T_s is the sampling period (=$1/F_s$), and α is the cyclic frequency to be detected. Note that the conjugate CAC is used to detect cyclic frequencies

Table 3.2: Cyclic features for some modulation classes that occur for conjugate $(\cdot)^*$ and non-conjugate CAC.

Modulation	Peaks at (α,ν)
PSK, QAM	$(\frac{1}{T},0)^*$
ASK	$(\frac{1}{T},0)^*$, $(2f_c,0)$, $(2f_c \pm \frac{1}{T},0)$
GMSK	$(\frac{1}{T},0)^*$, $(2f_c \pm \frac{1}{2T},0)$

close to baseband, whereas the non-conjugate CAC is used to detect the cyclostationary features at cyclic frequency α related to F_c. Different modulation classes can be differentiated via the cyclostationarity test because their CACs possess cyclic peaks at different locations of cyclic frequencies α, which is a function of F_s and F_c. Table 3.2 summarizes the locations of spectral peaks for the three targeted modulation classes in this work.

However, in blind classification scenarios, the estimated cyclic frequencies might not be equal to true cyclic frequencies. It was shown in [50] that computing the CAC at $\hat{\alpha} = (1+\Delta_\alpha)\alpha$, where α is the true cyclic frequency and Δ_α is the cyclic frequency offset (CFO), results in performance degradation in terms of the classification accuracy. The relation between the CAC at $\hat{\alpha}$ and that at α is given by

$$|\tilde{R}_x^{\hat{\alpha}}(0)| = |\tilde{R}_x^{\alpha}(0)| \times \left| \frac{\sin(\pi\alpha N T_s \Delta_\alpha)}{N \sin(\pi\alpha T_s \Delta_\alpha)} \right|. \tag{3.4}$$

Therefore, under a non-zero Δ_α, increasing the number of samples (N) does not improve the detection accuracy but instead degrade the cycliclostationary feature. This in turn respond to the aforementioned idea at the beginning of this subsection that we

need accurate parameter estimation to minimize the CFO and improve the classification accuracy.

With respect to the symbol rate estimation, we note from Table 3.2 that all SC modulation classes exhibit a cyclostationary feature at $\alpha = 1/T$. Therefore, detecting the presence of this cyclostationary feature would inherently estimate the symbol rate of the signal. The coarse and fine estimates of the symbol rate from BSG can be used to set the search window $\mathcal{W}_T$, within which the cyclic peak at the symbol rate will be located. The detection of the cyclostationary feature at $1/T$ is therefore obtained by solving the following optimization problem:

$$\max_{\alpha_i \in \mathcal{W}_T} \left| \sum_{n=0}^{N_T-1} |x[n]|^2 e^{-j2\pi\alpha_i n T_s} \right|, \tag{3.5}$$

where N_T is the number of samples per CAC computation used to estimate the signal's symbol rate.

Given that not all classes have the cyclostationary feature related to their carrier frequency, the CACs given in Eq. (3.2) and (3.3) cannot be directly used. The residual F_c estimation can be performed by detecting the cyclic feature at $\alpha = 4f_c$ after squaring the incoming samples [51] . We denote the search window by $\mathcal{W}_f$ within which the cyclic peak at $4f_c$ occurs. The estimation is therefore obtained by solving the following optimization problem:

$$\max_{\alpha_i \in \mathcal{W}_f} \left| \sum_{n=0}^{N_f-1} x[n]^4 e^{-j2\pi\alpha_i n T_s} \right|, \tag{3.6}$$

where N_f is the total number of samples per CAC computation used to estimate the signal's F_c. Note that with increasing number of samples over which the CAC is com-

puted, the noise is suppressed and the features of interest become prominent. As a result, both N_T and N_f are a function of the SNR of the received signal.

Solving the optimization equations (3.5) and (3.6) requires infinite computational complexity. As a result, the search space for the maximum cyclic feature has to be discretized. We denote by Δ_{α_T} and Δ_{α_f} the resolutions for the symbol rate and carrier frequency estimators. As a result, there are two degrees of freedom in the design of each of the algorithms: (1) the step size Δ_{α_T} and Δ_{α_f} within the window $\mathcal{W}_T$ and $\mathcal{W}_f$ respectively, and (2) the number of samples N_T and N_f required for the computation of every CAC at the cyclic frequency α_i of interest. The symbol rate and the carrier frequency estimation algorithms cannot yield estimation accuracies smaller than their respective step size Δ_{α_T} and Δ_{α_f}.

Also, the number of CAC computations required in Eq. (3.5) and Eq. (3.6) are equal to the cardinality of the discretized search windows $S_T = \lceil \mathcal{W}_T/\Delta_{\alpha_T} \rceil$ and $S_f = \lceil \mathcal{W}_f/\Delta_{\alpha_f} \rceil$ respectively. Given that both estimators use same CAC algorithm, their consumed energy per sample is pretty much the same. The only difference is from the energy consumed for squaring the input data, which is negligible compared to the entire CAC function. As a result, the total consumed energy of the *pre-processors* is proportional to $(S_T N_T + S_f N_f) T_s$. The choice of the design parameters (Δ_{α_T}, Δ_{α_f}, N_T, N_f) and their relationship to the required classification accuracy is explained in Section 3.3.1.

3.2.3 Modulation-Type Classifier

After the residual estimation, the modulation-type classifier computes the CAC at cyclic frequencies within the union of possible cyclostationary features in Table 3.2, resulting in a six-dimensional feature vector [52] given by

$$\mathbf{F} = \Big[|\tilde{R}_{x^*}^{1/T}(0)|, |\tilde{R}_x^{2f_c-1/T}(0)|, |\tilde{R}_x^{2f_c-1/2T}(0)|, \\ |\tilde{R}_x^{2f_c}(0)|, |\tilde{R}_x^{2f_c+1/2T}(0)|, |\tilde{R}_x^{2f_c+1/T}(0)| \Big]. \tag{3.7}$$

Because each element in the feature vector $\mathbf{F}$ is proportional to the received signal power, we normalize the feature vector to unit power, and compare this normalized feature vector $\bar{\mathbf{F}}$ to asymptotic normalized feature vectors $\bar{\mathbf{V}}_i$, $i \in [\text{QAM}, \text{PSK}, \text{ASK}, \text{MSK}]$, for each of the classes considered. For instance, the normalized asymptotic feature vector for signals belonging to QAM signal is $\bar{\mathbf{V}}_{\text{QAM}} = [1,0,0,0,0,0]$.

The resulting normalized feature vector is compared to each feature vector $\bar{\mathbf{V}}_i$, and the classifier picks the modulation class $\hat{C}$ whose feature vector is closest to one of the received signal in the least square sense, namely

$$\hat{C} = \arg \min_{i \in [\text{QAM,PSK,ASK,MSK}]} ||\bar{\mathbf{F}} - \bar{\mathbf{V}}_i||^2. \tag{3.8}$$

In contrast to the residual estimation, the only degree of freedom in the design of the modulation type classifier is the number of samples N_c required to compute each of the six CACs that form the feature vector. Given the SNR of the received signal and the accuracies of residual estimation, N_c is chosen accordingly to meet the desired classification probability. As a result of the six CACs required for classification, the

processing time for modulation-type estimation is equal to $6N_cT_s$. The six CACs are computed sequentially to enable high degree of hardware reuse without violating the processing time budget and compromising the total energy consumption.

3.2.4 Spread Spectrum Classification

Within the SC class, we distinguish between BPSK and direct sequence spread spectrum (DSSS) signals based on the variance $\rho(\tau)$ of the signal's autocorrelation at a given lag τ [53]. The received signal is divided into non-overlapping windows of N_d samples each. For each window, we compute the estimate of the autocorrelation for the possible expected lags. The fluctuations of the autocorrelation value for each τ of interest are then measured over M_d windows. It was shown [53] that these fluctuations have peaks at a lag equal to the code length. The algorithm has further been optimized to only search for code lengths that are a power of two as these are most commonly used. With this approach, the presence of a DSSS signal as well as its code length can be determined in a single step.

Mathematically, the autocorrelation function is approximated using N_d samples over all lags of interest $\tau \in 2^{[1,\ldots,6]}$ for each frame $m \in [1,\ldots,M_d]$ of input samples $x_m[\cdot]$, resulting in

$$r_x(m,\tau) = \frac{1}{N_d} \sum_{n=1}^{N_d} x_m[n] x_m[n-\tau]. \tag{3.9}$$

The variance of the autocorrelation function is computed at every lag given M_d realiza-

tions of $r_x(m,\tau)$

$$\rho(\tau) = \frac{1}{M_d}\sum_{m=1}^{M_d} r_x(m,\tau)^2 - \left(\frac{1}{M_d}\sum_{m=1}^{M_d} r_x(m,\tau)\right)^2. \tag{3.10}$$

Finally, in order to detect if the received signal is a spread spectrum signal with code length τ, the statistic $\rho(\tau)$ is compared to a SNR-dependent threshold γ_d. For example, $N_d = 32$ samples per frame and $M_d = 4$ averages are required for each lag τ to meet the 95% classification probability at 10dB SNR.

3.3 Energy-Efficient Processing of FEX Engine

This section presents an algorithm-architecture co-design methodology to make the FEX engine perform various classification tasks yet still achieve high energy efficiency. We firstly discuss the workload distribution among the aforementioned classification algorithms to define an energy-efficient programming model. Based on the analysis result, the proper core granularity of the FEX engine can also be determined. At the circuit level, we adopt a 1/2/4/8$\times$ programmable parallelism and the off-chip frequency-voltage scaling to the FEX engine for high-efficiency processing.

3.3.1 Processing Time and Energy Minimization: Algorithmic Perspectives

To minimize the consumed energy, we categorize the signal processing blocks into *dependent* blocks, whose design variables are a function of the output of previous blocks, and *independent* blocks, whose design variables can be independently determined (Fig. 3.4).

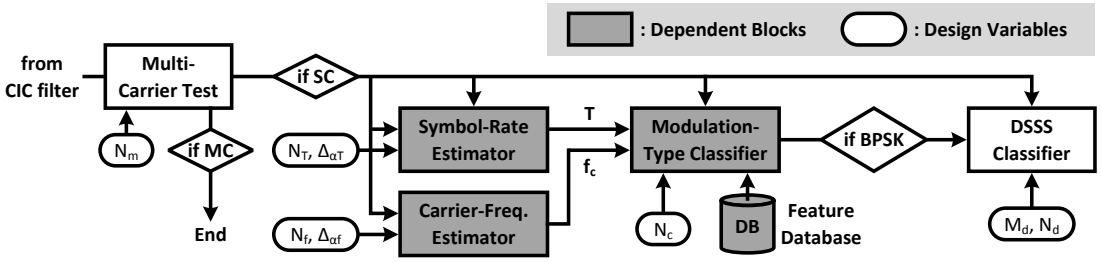


Figure 3.4: Dependent blocks (in gray) and their design variables to be optimized.

Specifically, the design variables of both the OFDM and DSSS classifiers are unrelated to any other stage of the classification, and are therefore labeled as *independent* blocks. The number of samples N_c spent on the modulation-type classification, however, is tightly related to the accuracy of the residual F_s/F_c estimation blocks. The *independent* blocks consume a fixed amount of energy regardless of the other blocks, and therefore are not jointly optimized with the rest of the blocks. On the other hand, a joint optimization of the total consumed energy of the *dependent* blocks is possible.

In order to optimize the energy consumption of the dependent blocks, we note that all of them use the CAC statistics in Eq. (3.2) and (3.3). Thus, minimizing the total number of samples spent for classification is fairly equivalent to minimizing the energy. Minimizing the total number of samples is also equivalent to minimizing the processing time $(6N_c + S_T N_T + S_f N_f)T_s$, where $S_T = \lceil \mathcal{W}_T/\Delta_{\alpha_T} \rceil$ and $S_f = \lceil \mathcal{W}_f/\Delta_{\alpha_f} \rceil$. The search windows $\mathcal{W}_T$ and $\mathcal{W}_f$ are obtained from the BSG and are SNR-dependent, and are therefore not optimized. Similarly, the number of samples per CAC computation N_T and N_f are also SNR-dependent since they are the minimum required number of samples to meet the detection probability. For instance, $N_T = N_f = 320$ samples at

10dB SNR. The only variables to optimize, as a result, are N_c, S_T, and S_f, which are equivalent to optimizing over N_c, Δ_{α_T}, and Δ_{α_f}:

$$\begin{aligned} &\min_{N_c, S_T, S_f} \quad 6N_c + S_T N_T + S_f N_f \\ &\text{such that } \mathcal{P}(\hat{C} = i \mid \Delta_{\alpha_f}, \Delta_{\alpha_T}, N_c, C = i) \geq 0.95 \\ &\qquad \forall i \in [\text{QAM}, \text{PSK}, \text{ASK}, \text{MSK}]. \end{aligned} \tag{3.11}$$

Note that the result of the optimization problem (3.11) is a function of the coarse estimate windows $\mathcal{W}_T$ and $\mathcal{W}_f$. In fact, the wider the windows are, the larger the number of CAC computations S_T and S_f are required for a given step size Δ_{α_T} and Δ_{α_f}, respectively. Therefore, the optimum choice of the design variables is related to the estimation accuracy from the BSG.

There exists an inherent tradeoff between the accuracies of residual parameter estimation and the modulation-type classification. As specified in Eq. (3.4) and approved by the simulation, the classification accuracy of QAM signals[1] is below the desired probability of 95% when $\Delta_{\alpha_T} \geq$1000ppm at 10dB SNR, even if we increase the number of samples. An intuitive explanation to this phenomenon is that, increasing the number of samples is only effective to the suppression of the noise, but doesn't help with the cyclic frequency offset. We refer to this SNR-dependent, maximally tolerable cyclic frequency offset as $\Delta_{\max \alpha_T}$, which equals to, for example, 1000ppm at 10dB SNR.

[1]We select the QAM signals to determine the maximum tolerable Δ_{α_T} because they only exhibit a cyclostationary feature at the F_s. Since the feature at F_s is the weakest among all cyclostationary features [54], it requires the most number of samples to meet the desired classification probability.

The accuracy of the carrier frequency estimation error Δ_{α_f} is determined by the SC signals that exhibit a cyclostationary feature at the carrier frequency. However, unlike the accuracy requirement for the symbol-rate estimation, Δ_{α_f} has to be jointly determined for every $\Delta_{\alpha_T} \leq \Delta_{\max \alpha_T}$. As a result, for every $\Delta_{\alpha_T} \leq \Delta_{\max \alpha_T}$ that guarantees proper classification of QAM signals, there exists a maximum estimation error $\Delta_{\max \alpha_f}$ that can be tolerated by the rest of the signal classes. To understand the tradeoff, we obtain the feasible region in the $(\Delta_{\alpha_T}, \Delta_{\alpha_f})$ coordinate system and formulate the following optimization problem:

$$\begin{aligned} &(\Delta_{\max \alpha_f} \mid N_c, \Delta_{\alpha_T}) = \max \ \ \Delta_{\alpha_f} \\ &\text{such that } \ \mathcal{P}(\hat{C} = i \mid \Delta_{\alpha_f}, \Delta_{\alpha_T}, N_c, C = i) \geq 0.95, \end{aligned} \tag{3.12}$$

where C is the correct class to which the received signal belongs to, and $i \in [\text{PSK}, \text{ASK}, \text{MSK}]$. Therefore, for every $\Delta_{\alpha_T} \leq \Delta_{\max \alpha_T}$, there exists a maximum $\Delta_{\max \alpha_f}$ under which classification requirement of 95% is met.

This tradeoff among different set of triplets is illustrated in Fig. 3.5 for a 10dB SNR. We conclude that that setting a stricter requirement on the symbol-rate estimator relaxes the required accuracy of the carrier frequency estimator. However, spending too much time (hence energy) on the symbol rate estimator to push Δ_{α_T} below 700ppm does not result in better relaxation for the carrier frequency estimator. This is because the cyclostationary features at a function of the carrier frequency cannot be detected reliably with an offset larger than 5400ppm at SNR of 10dB. Jointly considering these limitations gives us a *feasible* region to the optimization problem. As a result, although

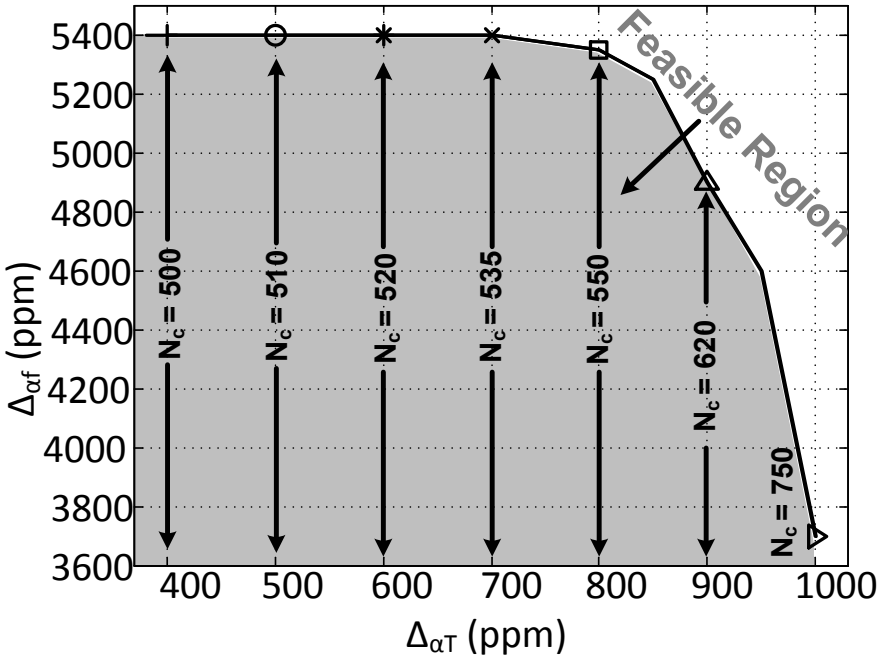


Figure 3.5: Tradeoff between the number of samples for symbol rate estimator and carrier frequency estimator at 10dB with 95% classification probability

there exists an infinite number of $(\Delta_{\alpha_T}, \Delta_{\alpha_f}, N_c)$ triplets that meet the required classification probability, the most energy-efficient triplets lie on the boundary of the feasible region. These SNR-dependent, energy-optimal combinations of design variables can be computed off-line and used to configure the FEX engine.

3.3.2 Algorithm-Architecture Co-Design

The algorithm-architecture co-design methodology is applied to implement the FEX engine, as illustrated in Fig. 3.6. Table 3.3 summarizes the list of algorithms and the associated workload partitions. Note that the numbers inside the parenthesis represent the workload *along* and *across*) the classification tasks (denoted as (*along*||b*across*) in

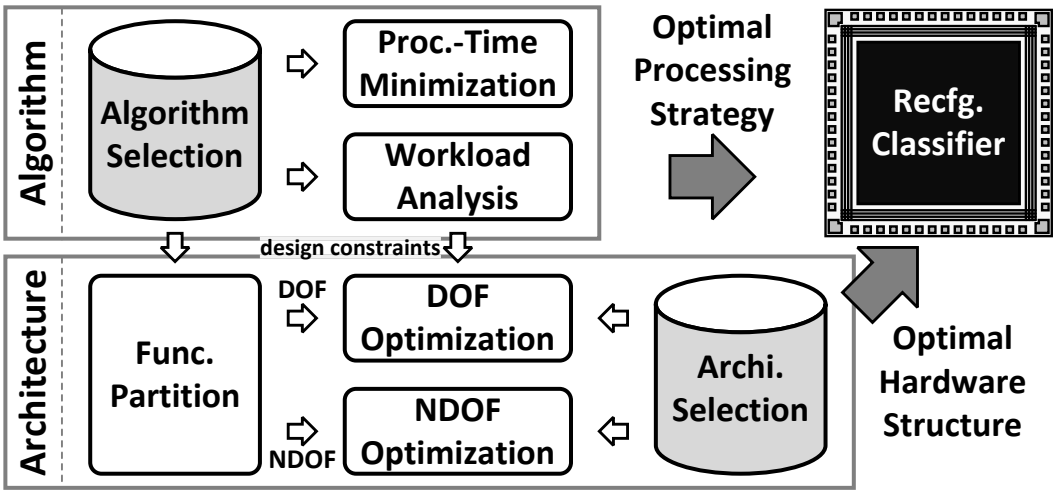


Figure 3.6: Algorithm-architecture co-design framework delivers optimized hardware as well as processing strategies.

% in the table). The classification algorithms are chosen for their algorithmic robustness, good classification accuracy and architectural similarity to enable high degree of hardware reuse. As a result, the reconstructed signals after BSG all undergo the variants of the complex multiplication-and-accumulation (MAC) followed by a magnitude computation, and only the post-processing on the computed magnitude is algorithm dependent. For instance, the CAC for the residual F_c/F_s estimation simply performs the *argmax* function that chooses the cyclic frequency to maximize the objective function, while the CAC for modulation-type classification needs Euclidean distance calculation and *argmin* to detect the signal class whose theoretical feature vector is closest to the computed feature vector.

The selection of algorithms directly affects the implementation strategy. From functionality perspective, the implementation can be partitioned into two parts. We call the

Table 3.3: Classification algorithms and workload partitions at 10dB SNR.

Task Partition of Processing Blocks

Task	MAA & MCU	PPU
OFDM	$C_{42} = \frac{1}{N_m}\sum_{n=1}^{N_m} \lvert x[n]\rvert^4 - \lvert C_{20}\rvert^2 - 2C_{21}^2$	$C_{42} \gtrless$ Threshold
F_c Est.	$\lvert R_{x^2}(\alpha_i)\rvert = \left\lvert 1/N_f \sum_{n=0}^{N_f-1} x[n]^4 e^{-j2\pi\alpha_i n T_s}\right\rvert$	$\max_{\alpha_i} \lvert R_{x^2}(\alpha_i)\rvert$
F_s Est.	$\lvert R_{x^*}(\alpha_i)\rvert = \left\lvert 1/N_T \sum_{n=0}^{N_T-1} \lvert x[n]\rvert^2 e^{-j2\pi\alpha_i n T_s}\right\rvert$	$\max_{\alpha_i} \lvert R_{x^*}(\alpha_i)\rvert$
Mod. Type	$\lvert R_x(\alpha_i)\rvert = \left\lvert 1/N_c \sum_{n=0}^{N_c-1} x[n]^2 e^{-j2\pi\alpha_i n T_s}\right\rvert$	Euc. Dist. + argmin
DSSS	$r_x(m,\tau) = 1/N_d \lvert \sum_{n=1}^{N_d} x_m[n] x_m[n-\tau]\rvert$	$\max_\tau \rho(\tau)$

Total Number of Samples ((*along*$\|$*across*) the task in %)

Task	MAA	MCU & PPU
OFDM	300 (97.7$|$0.05)	7 (2.3$|$0.10)
F_c Est.	401k (98.8$|$77.0)	5024 (1.2$|$75.6)
F_s Est.	116k (98.8$|$22.3)	1456 (1.2$|$21.9)
Mod. Type	3k (95.3$|$0.60)	148 (4.7$|$2.20)
DSSS	230 (94.3$|$0.05)	14 (5.7$|$0.20)

first part the degree-of-freedom (DOF) operation, meaning that this type of computation is required by all algorithms. The second part is the non-degree-of-freedom (NDOF) operation, whose hardware cannot be efficiently shared by different algorithms. In this sense, the multi-algorithm-accelerator (MAA) and the magnitude computation unit (MCU) are categorized as DOF, while the post-processing unit (PPU) is viewed as NDOF.

Another aspect of algorithm-architecture trade-off is described by the workload requirements. Considering the processing time *along* an algorithm in Table 3.3, we can see the MAA is active for >95% of the time, while the MCU and the PPU only work for a few clock cycles, having very low utilization. On the other hand, if we focus on the workload requirements *across* the algorithms, we see the residual estimations and the modulation-type classification take up a majority of the time and energy (>99%). Since all three algorithms are realized by variants of CAC functions, the architecture for DOF operations has to be inclined to side with CACs, and relatively against other functions (e.g. C_{42}), to have strong connection to the energy minimization strategy in Sec. 3.3.1. Distinct hardware design constraints for each of these components are therefore derived. The MAA has to support high-throughput with minimized dynamic energy which can be accomplished by applying *parallelism* and *aggressive voltage scaling* at the circuit level. In addition, the MCU and the PPU need to have minimized leakage when staying idle due to their low utilization. Combined with the algorithm-level energy minimization strategy in Section 3.3.1, the entire co-design framework is able to deliver high energy efficiency from both hardware and software perspectives.

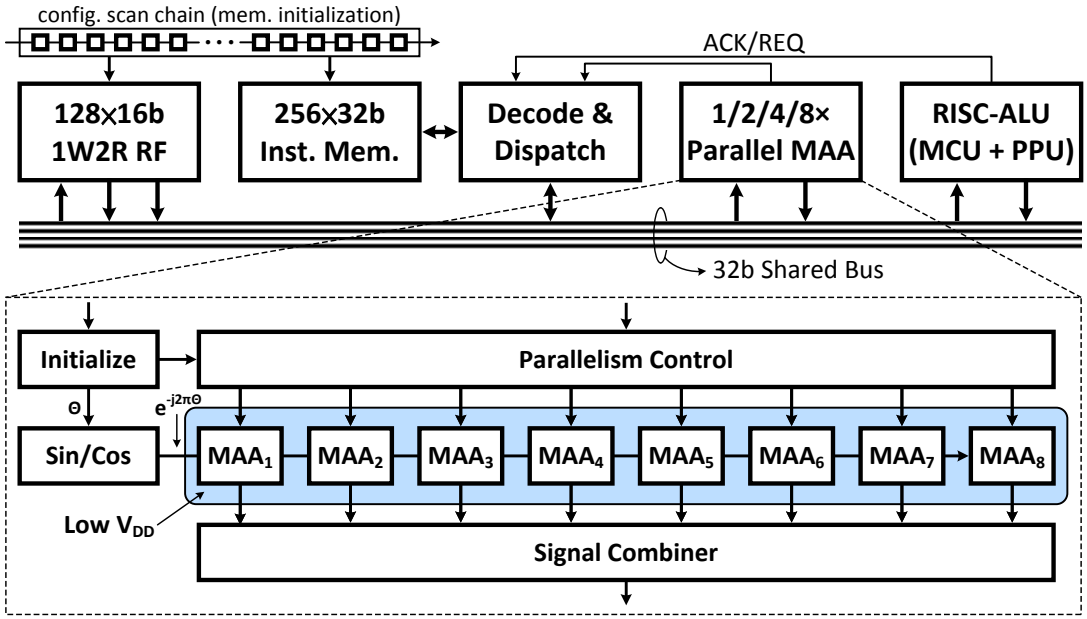


Figure 3.7: Feature extraction (FEX) engine for residual parameter estimation and modulation classification.

3.3.3 Proposed Architecture

The proposed FEX engine is a RISC-style processor with a custom 32b instruction set and domain-specific kernels (Fig. 3.7). It employs a 128×16b one-write-two-read (1W2R) register file (RF) for function variables (e.g. CAC frame length) and data access, and a shared 32b bus for data transfer between compute elements and external interfaces. The size of the 256×32b instruction memory is minimally required to map all of the tasks without requesting on-line reprogramming. The RISC-ALU consists of a PPU for generic operations (e.g. addition, shifting) and a CORDIC-based MCU for data rotation and vectoring. The MAA kernels handles domain-specific operations of classification tasks, and leverages programmable 1/2/4/8× parallelism with

scaled supply voltage (V_{DD}) for high energy efficiency. The request-acknowledgement protocol manages block synchronization and parallel processing to improve program runtime. Unlike traditional processors that unify their datapaths, the FEX engine is a hybrid-datapath system, doing complex-valued computation in MAA and MCU, but real-valued processing in PPU. Each processing block is individually optimized with particular design constraints derived from its workload requirements. The architectures of complex multipliers in MAA are carefully selected based on their propagation delay and area cost. Detailed implementation issues are presented as follows.

3.3.3.1 Multi-Algorithm Accelerator

Figure 3.8 shows the architecture of MAA, with its internal bit-width optimized by an in-house analysis tool [55]. The MAA is particularly dedicated to the critical operations of classification algorithms. It catches the complex-valued data ($x[n]$) directly from the outputs of BSG and passes them through a series of multipliers and/or squarers to generate their second- or fourth-order products. The products are then optionally passed through another complex multiplier (in CAC mode) before reaching the final accumulation stage. The formula C_{42} for OFDM classification is decomposed into three parts ($\frac{1}{N_m}\sum|x[n]|^4$, C_{20} and C_{21}), separately computed by MAA and stored in the system RF, and finally combined by the PPU. The two-mode squarer is flexible to perform either the squaring or the absolute-squaring of a complex number $a+jb$ efficiently by

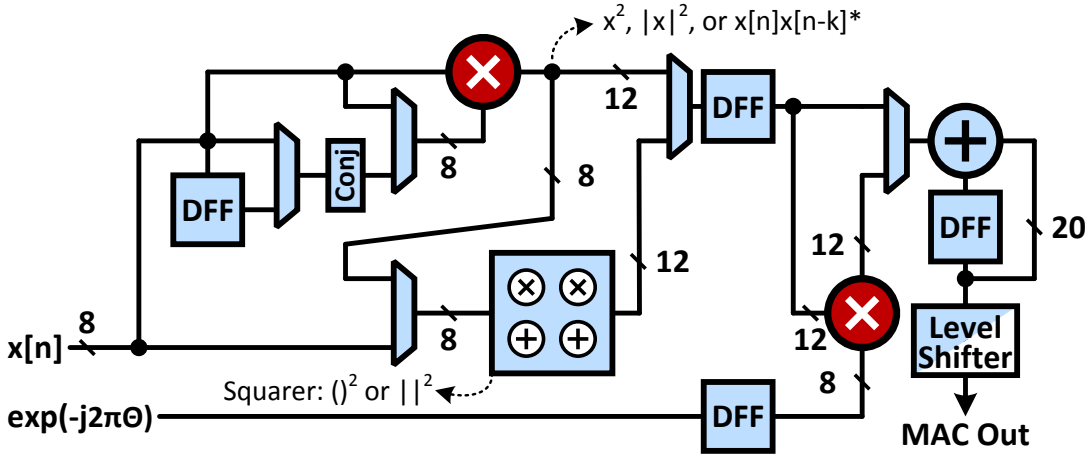


Figure 3.8: Multi-algorithm accelerator (MAA) unit. The DFF denotes the D-flip-flop, and the two multipliers highlighted in red represent the complex multipliers. The entire logic operations of MAA is done at low supply voltage (highlighted in light blue), and transformed to high-swing signaling in the end via the standard-V_t level shifter.

the following reformulation:

$$(a+jb)^2 = (a+b)(a-b) + j(2ab), \tag{3.13}$$

$$|a+jb|^2 = (a+b)(a+b) - (2ab). \tag{3.14}$$

Compared to the direct-mapping approach that requires three 8b multipliers and two 12b adders, the proposed method only uses two 8b multipliers, two 8b and one 12b adders, saving 28% of area.

The two complex multipliers in MAA are realized using the traditional four real multiplications and two additions ($4\times$, $2+$) rather than the method suggested in [57] that uses ($3\times$, $5+$) due to several reasons. Conventionally, trading one multiplier for three adders in the ($3\times$, $5+$) approach is beneficial since the complexity of multipliers is

usually much higher than that of adders for general-purpose processors. However, since the wordlength of complex multiplication in MAC is small, the original form is simpler. To see the tradeoff between (4×, 2+) and (3×, 5+) regarding their area estimates, we use the array-multiplier approximation for first-order comparison. Without loss of generality, the normalized size of an array multiplier can be estimated by the product of wordlengths of the multiplier and the multiplicand [58]. The area estimate of a (3×, 5+) complex multiplier is thus generalized by the following equation

$$Area_{3\times 5+} = 3L^2 + 10L, \tag{3.15}$$

where L denotes the wordlength. On the other hand, the area of a (4×, 2+) multiplier can be formulated as

$$Area_{4\times 2+} = 4L^2 + 4L. \tag{3.16}$$

Solving these two equations shows that (3×, 5+) can only be noticeably better (by 20%, for example) when the wordlengths of its operands are greater than 20 bits. In our case, these two candidates for an 8b multiplier realization only differs by 5.5%. The other concern to the argument is about the propagation delay. It is obvious that the (3×, 5+) approach is slower than (4×, 2+) due to the delay from an additional adder stage. As a consequence, the (4×, 2+) complex multiplier can use smaller logic gates to achieve the same delay as (3×, 5+), or it can exploit the advantageous timing slack to allow more voltage scaling, further minimizing its power consumption.

3.3.3.2 CAC Coefficient Generator

The CAC coefficient generator, illustrated as **Sin/Cos** in Fig. 3.7 and detailed in Fig. 3.9, generates complex exponential terms for CAC functions. It starts with a free-running angle accumulator whose step size equals the product of cyclic frequency and sampling rate ($\alpha_i T_s$). Note that the accumulator doesn't need to be reset before each CAC computation because any of its initial and common phase offset will be eventually eliminated through MCU. Following the accumulator is the angle synthesizer. It is realized in an area-efficient way by the piecewise-linear approximation method [56], plus a re-mapping circuit to generate sine/cosine values whose angles are out of the range between 0 and $\pi/4$. The area efficiency from the piecewise-linear approximator comes with the loss of accuracy. The synthesizer suffers a mean-square-error (MSE) of -40dB when it generates certain angles, meaning that it won't perform any better even in floating-point systems. However, such error is below the noise floor at $\geq$0dB SNR and therefore can barely affect the classification performance.

3.3.3.3 Magnitude Computation Unit

The CORDIC algorithm is implemented to compute the rotation and the scaling of a complex number. The core building block of a CORDIC consists of adders and shifters. The output precision depends on the number of CORDIC iteration stages N_i. There are three different types of architecture to implement CORDIC, i.e. fully pipelined, fully folded, and a hybrid between these two. Pipelined CORDIC achieves the highest

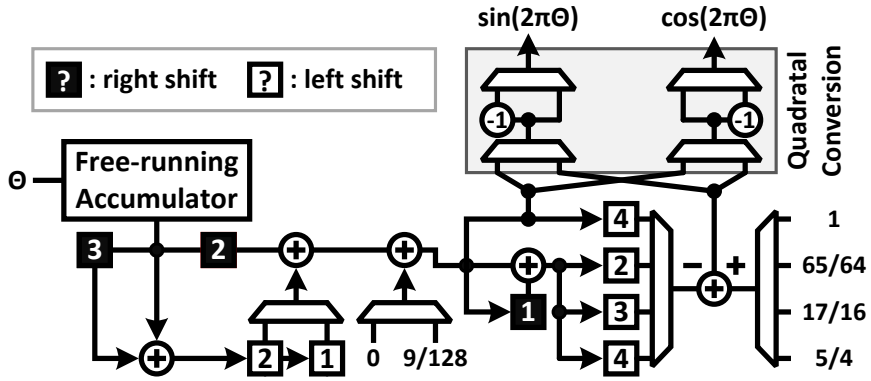


Figure 3.9: The coefficient generator provides the complex exponential terms for CAC by using only simple adders and shifters. The numbers inside the squares denote the amount of right or left shifting with sign extension.

throughput with high area and leakage penalty. The folded architecture takes N_i cycles to calculate the magnitude with around N_i-times lower area and leakage cost. Since the magnitude computation is highly underutilized and is only required at the end of each MAC tasks, the fully folded CORDIC architecture is implemented.

3.3.3.4 Post-Processing Unit

The PPU is a real-valued, one-cycle-latency arithmetic logic unit (ALU). It consists of a comparator (for threshold comparison), an 8-bit (8b) right/left-shifter (for power-of-two normalization), a 16b adder/subtractor, a 16b multiplier, and bit-wise operations such as bit-wise inversion, AND, OR, and exclusive-OR. For most of the time, PPU executes the normalization and/or the threshold comparison on the MCU outputs. The real-valued adder/subtractor and multiplier are occasionally used to compute the Euclidean distance required by the modulation-type classification. Instead of using a

divider to normalize the computed CAC feature vector, the multiplier is employed to de-normalize the theoretical feature vector before subtracting it by the computed one. The same multiplier is then reused to perform the squaring operation to complete the Euclidean distance calculation. The ALU operations are executed sequentially, one in each clock cycle, to realize the complex operations in an area-efficient way. Although the average operational latency from this approach is much longer than the one which does all operations in parallel, the cycle-time overhead is still negligible since the PPU is only active $<1\%$ of the total processing time. The slowest yet simplest PPU architecture minimizes the area and leakage.

3.3.3.5 Data Transfer and Control

The FEX engine employs a central controller to decode and deploy the control signals. The 32b instruction-set architecture (ISA) supports regular register-type instructions for the RISC-ALU and the RF, and special instructions for the MAA. The ISA also implements loop and jump instructions to efficiently exploit the memory space. The program counter continues to accumulate when it executes the regular one-cycle-latency instructions, but is halted during the long-latency MAA operations that usually takes hundreds to thousands of clock cycles.

All of the processing blocks use the simple request-acknowledge protocol to communicate with the central controller, telling the controller when to let the RF access their outputs. For illustration, the programming of MAA is depicted in Fig. 3.10. The controller first sends the request signal *REQ* and the initialization information *INIT*

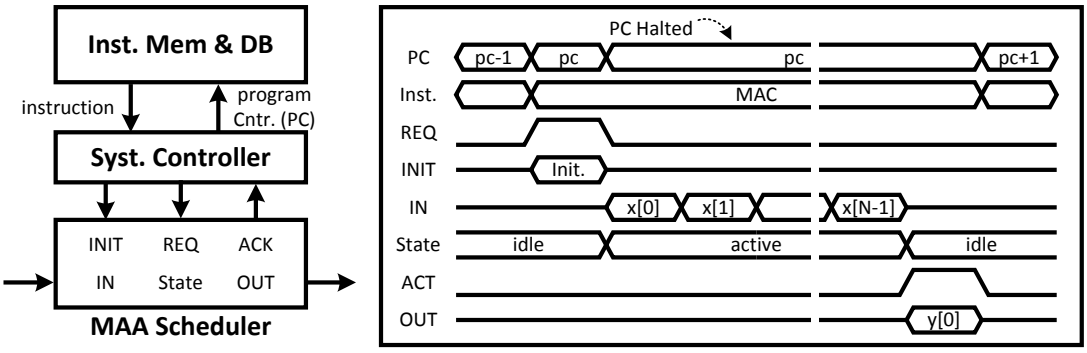


Figure 3.10: Programming of MAA unit via simple request-acknowledge protocol. The program counter (PC) is halted during MAA operations, and resumes to access the next instruction address after receiving the acknowledgement signal.

(e.g. the frame length and the level of parallelism based on the contents of the instruction) to activate the MAA. The MAA then starts the parallelism scheduling, the computation, the signal combining, and it generates an acknowledgement signal *ACK* along with the outputs when the task is finished. Upon catching the acknowledgement signal, the halted program counter resumes to access the new instruction, allowing the controller to store the MAA outputs into the RF for later processing. The MAA finally returns to the idle state, preparing to accept another request whenever needed. Since most of the processing time is spent on the MAA, the overhead from the data movement is negligible to the total processing time and energy. The instruction and memory overhead, however, cannot be ignored since they are active every cycle to control the processing blocks. The energy cost from the ISA and the processing blocks, as a result, are jointly considered during chip implementation.

Lastly, since the classification tasks are already determined at the co-design stage,

there is no need to use generic software compilers (e.g. C++) to program the processor. An assembly code, instead, is manually written and converted to machine code by an in-house assembler for design verification.

3.4 Chip Measurements

The BSG occupies 1.56mm^2 with two V_{DD}s for logic (0.65V) and memory (0.75V). The full-PSD scheme consumes 23.7μJ for a 500MS/s throughput, while the partial PSD dissipates an optimal 7.2μJ when a 10MHz signal is present, since only 1/16 of the 500MHz spectrum needs to be analyzed. The 3.3$\times$ (=23.7/7.2) of energy saving from measurements validates the accuracy of our tradeoff analysis (Fig. 3.11). The FEX occupies 0.13mm^2. Its 8$\times$ parallel MAA kernels are robust down to 0.35V at 25MHz for narrowband, and scale to 0.56V at 500MHz for wideband signal classification (Fig. 3.12), consuming 0.23mW and 7.1mW, respectively. The parallelism provides up to 2.2$\times$ of power reduction compared to a non-parallel MAA scheme where power varies from 0.3mW (0.42V, 25MHz) to 15.6mW (0.87V, 500 MHz). By including the high-V_{DD} FEX blocks, a peak efficiency of 5.6GOPS/mW (11.5pJ/sample) is observed at 100MHz, with 0.4V low and 0.55V high V_{DD} (limited by memory). At this minimum-energy point, in addition, the energy per computation is 1.6$\times$ lower compared to 18.4pJ/sample at 500MHz. An overall 3.1$\times$ efficiency improvement is achieved by the 8$\times$ parallelism and minimum-energy operation. By comparing the 5.6GOPS/mW with the theoretical upper bound of 50-55GOPS/mW in Chapter 2, we

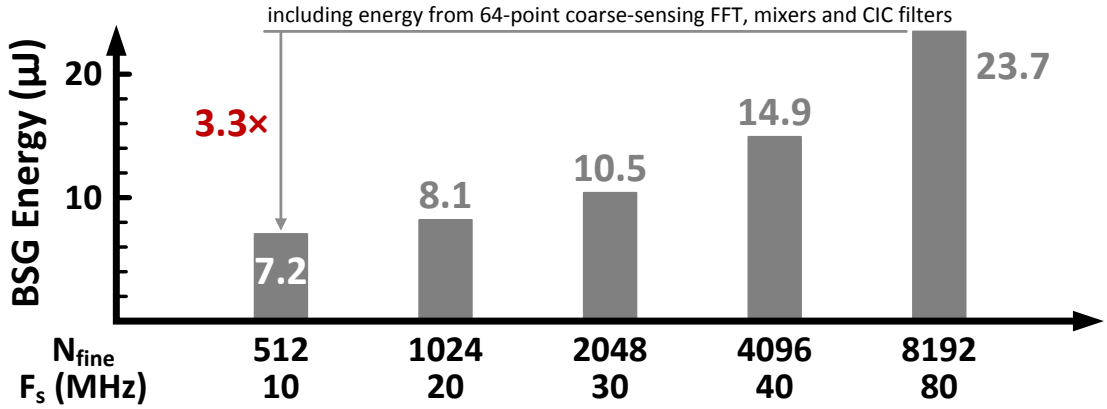


Figure 3.11: Measurement results of band segmentation engine shows a 3.3× of energy saving from partial-PSD sensing over the full-PSD case.

can fairly quantify the control and datapath overhead of FEX to be around 10×, which translates to an efficiency gap of 2.5-5× toward dedicated designs (whose overhead is generally 2-4×).

Combining both BSG and FEX, the processor consumes an average classification energy of 17μJ with <2ms processing time, a 59× energy reduction compared to an exhaustive search by FEX only (31× from the full PSD and another 1.9× from the partial PSD), while achieving 95% P_D and 0.5% P_{FA} at 10dB SNR (Fig. 3.13). Classification at 0dB SNR requires about 15× higher energy due to longer processing time, but the benefits over exhaustive approach still hold. The energy saving comes from the proposed three-step estimation, the tradeoff analysis of BSG configurations and the energy-efficient FEX implementation. The chip summary is shown in Fig. 3.14. A 10mW of average power while delivering 500MS/s qualifies the practicability of this processor for real-time blind classification in CRs.

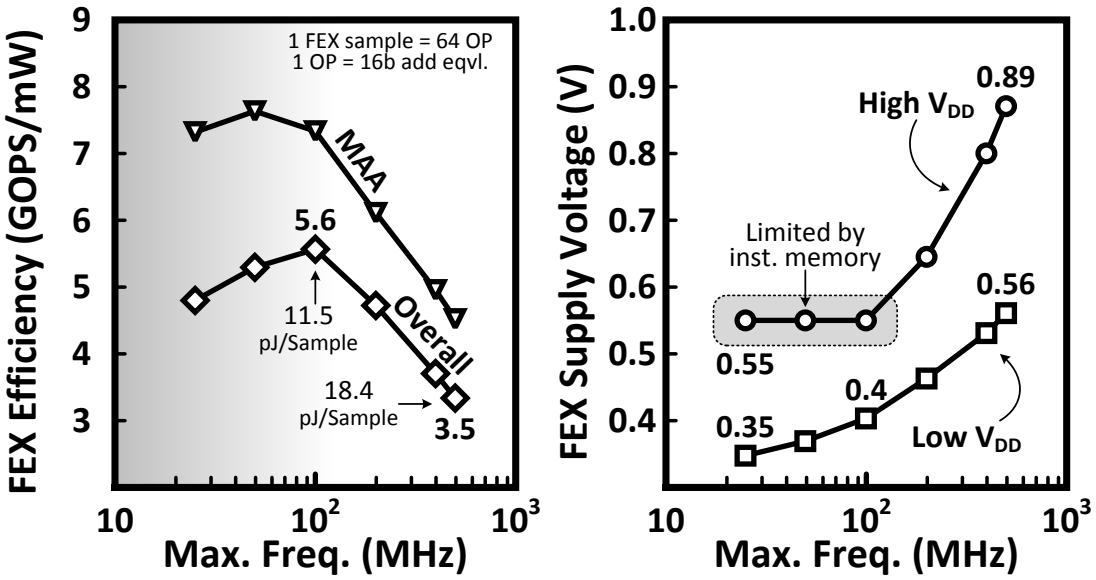


Figure 3.12: Measurement results of feature extraction engine shows a 3.1× of efficiency improvement by using the parallelism technique and operating the MAA kernels at minimum energy point.

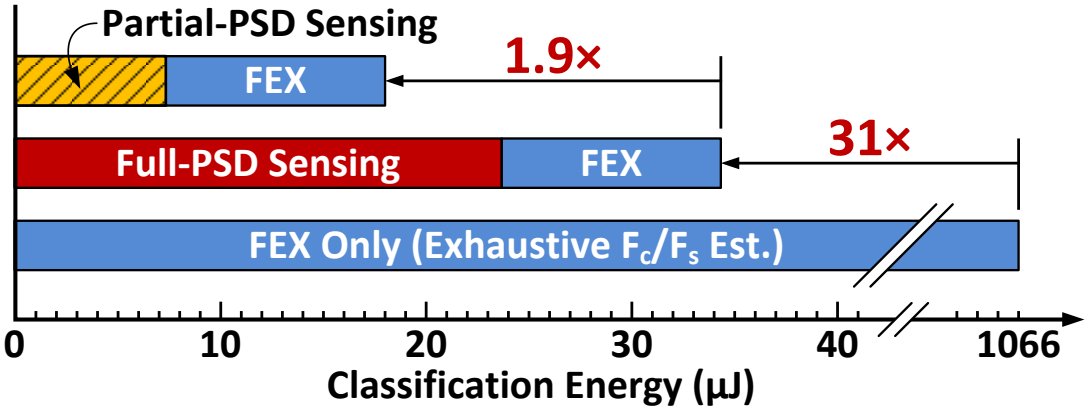
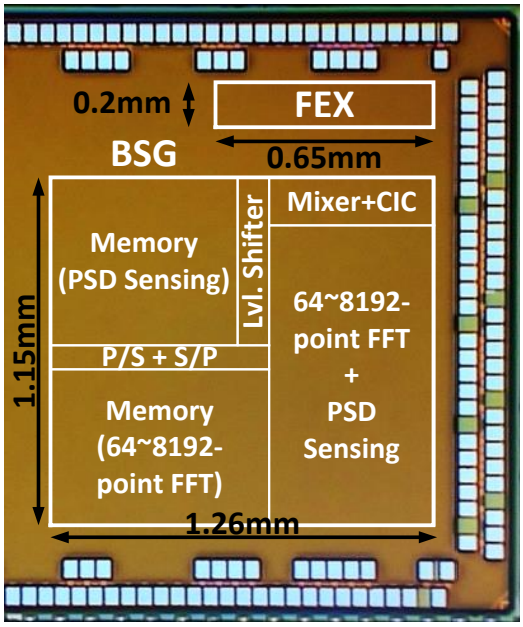


Figure 3.13: Energy breakdown of blind classification at 10dB SNR. A total of 59× energy saving compared to an exhaustive parameter estimation by FEX only (31× from the full PSD and another 1.9× from the partial PSD) is achieved.



Technology	TSMC 40nm CMOS	
Kernel	Band Segmt. (BSG)	Feature Ex. (FEX)
Area (mm^2)	1.45	0.13
Gate Count (NAND2 Eqvl.)	2.4M	0.25
Logic V_{DD} (Volt)	0.65	0.35-0.56
Mem. V_{DD} (Volt)	0.75	0.55-0.90
Throughput (MS/s)	500	25-500
Power (mW)	10.2	0.37-9.27
Peak Efficiency (GOPS/mW)	5.1	5.6

Figure 3.14: Chip micrograph and performance summary.

3.5 Further Improvement by Dynamic Resource Management

Although the first classification DSP successfully tackles the challenges of blindly estimating carrier frequency (F_c) and symbol rate (F_s) in real time, as well as building efficient hardware for multi-algorithm supports under a wideband channel, there remains one problem unresolved. The problem is how to keep the circuit's energy efficiency high regardless of the throughput requirements. To understand why this problem exists, and consequentially the motivation of the second chip, let's review the two things we learned from the first chip:

- The clock rate of FEX depends on the detected F_s from BSG. The higher the F_s is, the higher the throughput that FEX needs to provide.

- Parallelism and voltage scaling effectively improve the energy efficiency.

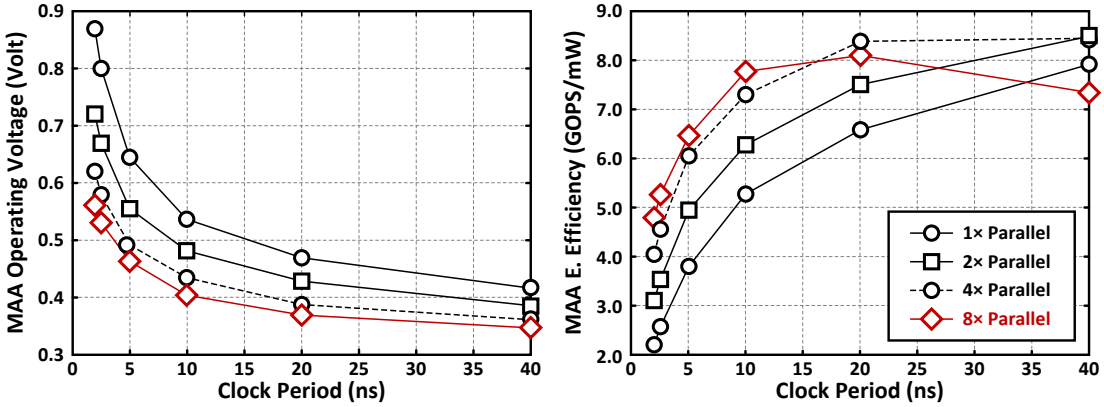


Figure 3.15: Voltage and efficiency plot of MAA kernels with different parallelism options. The optimal parallelism decision is dependent to the clock period and, implicitly, signal bandwidth.

However, as the F_s of the signal is totally unknown, we cannot assume any throughput requirement for the FEX in advance. Such variable throughput nature will impede the circuit from always achieving the peak energy efficiency, if it is unable to adjust the parallelism and the supply voltage in real time.

3.5.1 Dynamic Parallelism and Frequency Scaling

The red curve in Fig. 3.15 redraws the measurement results of the MAA kernels with 8× parallelism. The left-hand side shows the minimum voltage requirement versus the clock period, which is inversely proportional to the throughput and the detected F_s. The right-hand side shows the corresponding MAA energy efficiency. We can see from this plot that, although the MAA kernels are claimed to achieve a peak efficiency of around 8GOPS/mW, the efficiency curve actually varies by a lot, all the way from 8

down to 4.5GOPS/mW as a function of the clock period. In order to make the efficiency more invariant to the throughput changes, we need to look into the curves of other parallelism options ($1\times/2\times/4\times$). Note that because we didn't implement power gating on the first chip, we need to normalize the leakage contribution of each parallelism case for a fair comparison (i.e. to "pretend" as if there's power gating on each MAA core in the chip). For instance, the energy of a $2\times$-parallel MAA is calculated by summing up the switching and 1/4 of the total leakage energy, while for the $4\times$ case the leakage number becomes 1/2 of the total. The argument becomes more clear now. The $8\times$ is not always the best solution in energy efficiency. Instead, when the throughput gets lower, the $4\times$, $2\times$, or even don't do the parallelism will be the best strategy. So now, if we can efficiently hop around different parallelism options according to the F_s we detect in BSG, we will likely to keep the efficiency always close to the optimal value, as highlighted in Fig. 3.16. However, we see the supply voltages of different parallelism decisions also need to be slightly scaled to compensate the parallelism overhead (from the serial-parallel and parallel-serial circuits). But here we do propose to fix the voltage to certain safe value with some design margin due to the considerations regarding the process variation and the energy overhead from a highly-accurate on-chip voltage regulator. Specifically, as the MAA cores operate at near-threshold region, any small voltage fluctuation will impact the operating frequency by a lot. In order to reliably deliver the voltage, we must need a high-resolution regulator on the chip, whose energy overhead can easily offset the efficiency benefit from the accurate supply. As a result, we discard the use of such high-performance regulator but stick with a constant voltage

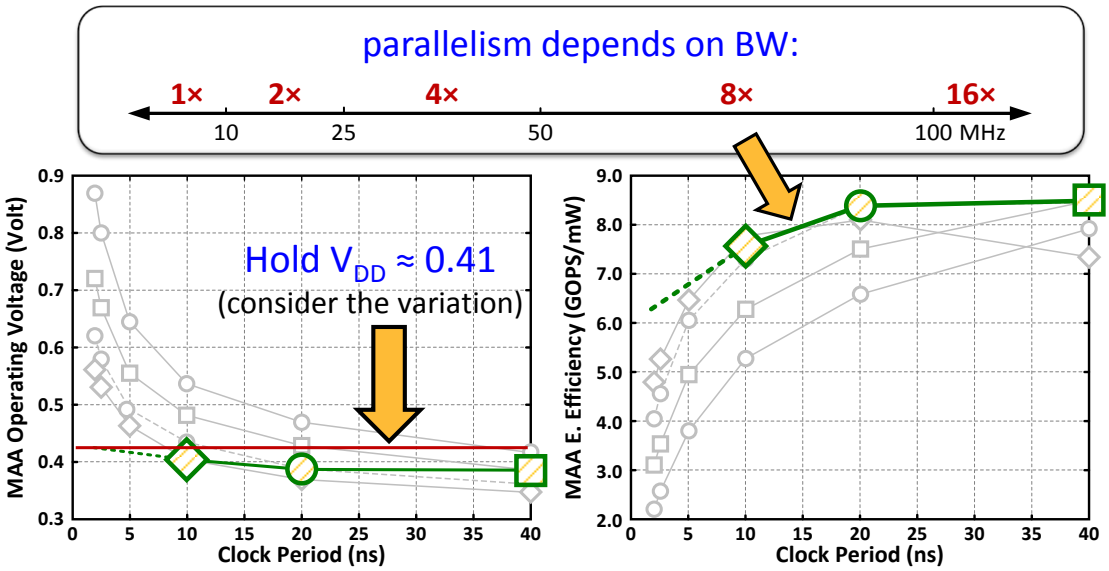


Figure 3.16: Dynamic parallelism and frequency scaling improves the energy efficiency. The voltage is proposed to stay constant by considering the process variation and the energy overhead from voltage regulator.

to simplify the chip design flow without losing too much efficiency (<5%) due to the deviation between the operating and the optimal supply value. The proposed concept of *dynamic parallelism and frequency scaling (DPFS)* thus requires a high-performance power gate on each MAA, as well as an on-chip scheduler to turn on/off the cores and adjust the clock rate dynamically.

3.5.2 Multi-signal Detection and Classification

Figure 3.17 shows layout view the second chip: a 500MHz wideband blind classification system-on-a-chip (SoC). In this new design, we embed the analog frontend that includes the time-interleaved ADCs and the analog FFT processor; the digital fron-

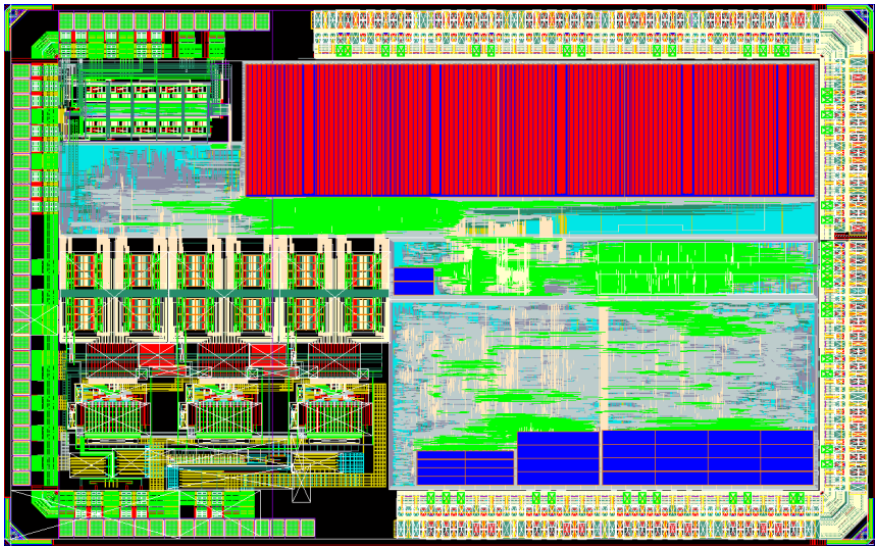


Figure 3.17: Top-level layout of the wideband classification SoC.

tend that includes the ADC calibration/compensation circuits and the analog-digital interfaces; the multi-signal classification DSP that includes a lot of innovation over the previous DSP chip.

As highlighted in Fig. 3.18, we modify the BSG to estimate up to 32 signals' frequency information per PSD sensing. We also have an on-chip clock divider to provide eight clock domains for the reconstructed signals. We have four mixer-filter pairs so that the new design can classify up to four signals at a time. Since each of the signals to be classified are completely independent, their symbol rates are unrelated as well. Therefore, we are ending up with taking care of four asynchronous signal threads with totally different clock rates for the classification. These asynchronous threads are going to share the 16 MAA kernels for parallel processing with the help of our proposed multi-core scheduler, parallelism mapper, and signal combiner.

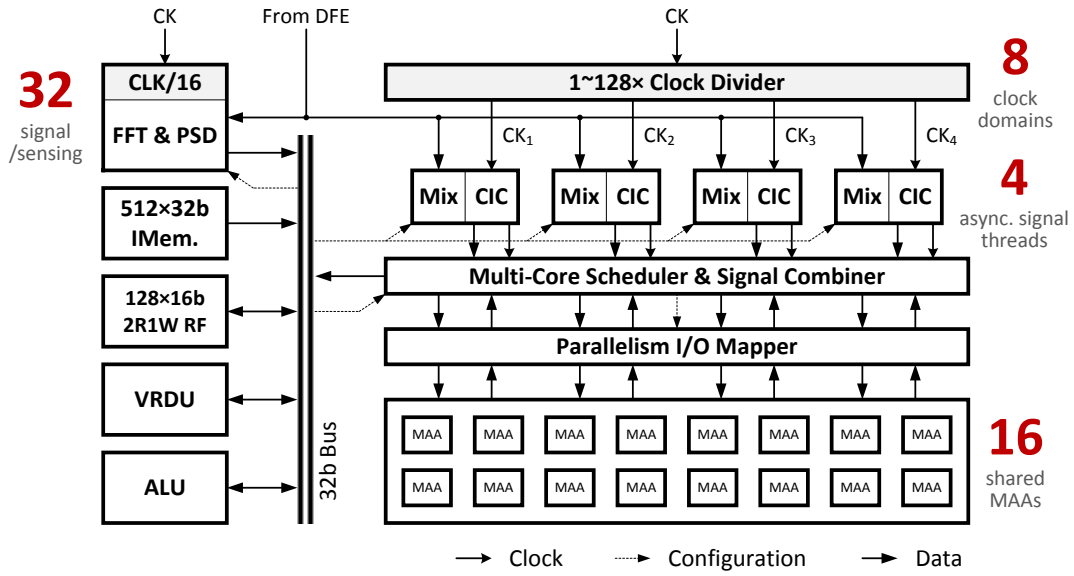


Figure 3.18: Architecture of the multi-signal classification DSP.

3.5.3 Multi-core Scheduling

As DPFS improves the system's energy efficiency by shutting down the under-utilized MAAs, we can also handle the problem in an opposite way by making all MAAs *busy* to improve the hardware utilization. Higher hardware utilization with proper core scheduling can lead to shorter processing time and energy cost per signal thread. In our new design, we embed a multi-core scheduler that manage the requests from the four signal threads on a first-come-first-serve basis. The original serial-to-parallel (S2P) and parallel-to-serial (P2S) circuits are replaced by the network-like mapper and de-mapper, respectively. Since the threads are totally asynchronous, we need to make the necessary adjustment by sending not only the data but also the clock signal of the thread to the corresponding MAAs assigned by the scheduler. As the scheduling is totally dynamic, the fixed relation between the time elapse and the S2P/P2S datapath as

	Chip 1	Chip 2	
Technology	40nm Regular V_{th}	40nm High V_{th}	
Analog	No	**AFFT + ADC**	**Complete Solution**
Area	1.58 mm^2	3.35 mm^2	
Channel	AWGN, 500MHz	**Multipath**, 500MHz	**Realistic Environment**
PSD Capability	1 Signal/Sensing	**32** Signal/Sensing	**Energy Efficiency**
Classifi. Thread	1	**1~4**	**Versatility**
On-chip Clocking	No	**Yes**	
Dync. Scheduling	No	**Yes**	
Power Gating	No	**Yes**	
Peak Throughput	500MS/s	**600MS/s**	
Peak BSG Effi. (GOPS/mW)	5.1	**8.0 (est.)**	**Lower leakage @E_{min}**
MAA Efficiency (GOPS/mW)	4.5~7.8	**11.9~13.6 (est.)**	**Higher efficiency & more invariant to F_s**
Processing Time per Signal @10dB	2ms	**0.5ms (est.)**	**Lower proc. time & higher HW utilization**

Table 3.4: Performance comparisons show the efficiency and flexibility benefits of the second over the first classification IC.

seen in traditional parallelism circuits no longer holds. Instead, the thread won't know which MAAs provide the services, nor can it specifically choose any MAA resource – Everything is decided by the scheduler.

3.5.4 Projected Efficiency

Table 3.4 shows the comparison between the first and the second classification chip. We deliver a more complete solution by integrating analog and digital parts. By supporting new algorithms, the new chip is able to handle multipath channel environment.

We achieve higher BSG energy efficiency from multi-signal (up to 32) frequency estimation. The chip is also more versatile as its parallelism, frequency and resources are dynamically managed. The use of high-V_th devices helps to achieve lower leakage at minimum-energy point. Since the switching energy is known to be roughly equal to the leakage at minimum-energy point, we can expect higher peak energy efficiency from the new design. Lastly, and most importantly, the FEX efficiency is expected to become not only higher, but also more invariant to the symbol rate due to DPFS techniques. The multi-core scheduling helps to reduce the processing time per signal by improving the hardware utilization.

3.6 Chapter Summary

This chapter presents the wideband blind classification processor for cognitive-ratio networks.

The proposed chip operates at $\geq$0dB SNR in a 500MHz channel, targeting a $\geq$95% detection probability and a $\leq$0.5% falst-alarm rate with a 62.5kHz frequency resolution. It supports three-step (coarse-fine-residual) carrier-frequency (F_c) and symbol-rate (F_s) estimation for high energy efficiency and low processing time. The unknown signal is located, down-converted and down-sampled in the coarse and fine steps by inferring its power spectral density. A cyclic-autocorrelation function analyzes the reconstructed signal to achieve $<$1000ppm residual frequency error. Down-sampling enables 5-50$\times$ lower clock rate and 30-40% lower voltage in the residual estimation, for up to 1.6$\times$

higher energy efficiency. The average 76× processing time reduction compared with the CAC-based exhaustive search method further saves the overall classification energy. The key processing blocks are: (1) Two-level hierarchical, FFT-based band segmentation engine (BSG) for coarse and fine F_c/F_s detection and signal reconstruction; (2) Feature extraction processor (FEX) for residual frequency estimation and signal classification. Five modulation schemes can be classified: multicarrier, single-carrier PSK/QAM/MSK and spread spectrum. The BSG and the FEX each has two voltage domains for logic and memory to facilitate energy minimization based on the evolving down-sampling results. Compared to an exhaustive method, our designs achieved a 59× saving in classification energy. The energy saving comes from the proposed three-step estimation, the tradeoff analysis of BSG configurations (3.3× from full- to partial-PSD sensing) and the energy-efficient FEX implementation (3.1× from parallelism and voltage scaling). Overall, the chip consumes 17μJ within 2ms sensing time per classification at 10dB SNR.

Apart from the first classification DSP, we also introduce the concept of DPFS and the multi-core scheduling to not only improve the efficiency but also make it more invariant to throughput changes. The new classification SoC is believed to provide a more complete solution (analog plus digital) with multi-signal classification capability, further improving the circuit's efficiency and flexibility.

CHAPTER 4

Design Example 2: A 13.1GOPS/mW 16-Core Baseband Processor

This chapter demonstrates a 16-core processor for software-defined radios in 40nm CMOS. Featuring domain-specific kernels, flexible control and multi-scale interconnects, the processor achieves a peak energy efficiency of 13.1GOPS/mW (76fJ/OP) at 415mV, 25MHz, and a peak performance of 1.17TOPS at 1V, 500MHz, showing $>2.4\times$ higher energy efficiency than state-of-the-art communication chip multiprocessors, and closing the gap with functionally-equivalent ASICs to within $2.6\times$.

4.1 Existing Work and Problem Statements

As seen in the introduction chapter, there exists an inherent tradeoff between the flexibility and efficiency. Traditional ways of designing circuits, as a result, can hardly balance the two criteria. Instead, today's designers have to *hybridize* the design concepts between dedicated hardware and programmable processors to break the tradeoff.

The domain-specific reconfigurable processor (DSRP) is so far the most promising solution to balance the two design criteria [16]. It offers fairly enough flexibil-

ity and near-ASIC efficiency by spatially mapping the targeted set of algorithms to a unified architecture (usually an array of computing elements linked by on-chip interconnects), reducing the control overhead associated with programmable DSPs and general-purpose processors. Architectures of domain-specific processors have been extensively studied with emphasis on the core granularity [59] [60] [61] and the on-chip interconnects topology [62] [63] [64] [65]. Advanced circuit techniques such as low-swing interconnect signaling [66], distributed dynamic voltage and frequency scaling (DVFS) [65], fine-grained power gating [65], and globally-asynchronous-locally-synchronous (GALS) clocking [63] have also been used to enhance the energy efficiency of the processors. However, the benefits of control circuit simplifications for instruction storage, fetching and decoding are not thoroughly investigated. Prior work with single- or multiple-instruction-multiple-data (SIMD or MIMD) instruction set reduced the control overhead by exploiting data- or task-level parallelism. These solutions worked by complying with the signal processing characteristics of targeted application domains, but their instruction set architecture (ISA) had to be *predefined* in hardware and couldn't be changed. Such inflexible ISA causes problems with complexity, performance, and efficiency on domain-specific processors. In addition, the fetching and decoding circuit has to be complex to make the inflexible ISA universal enough to tackle the required flexibility, thereby increasing the area and energy overhead. Even worse, the design volatility (especially for the evolving wireless standards) may create a need for new instructions, but a non-expandable ISA on the same processor might either fail to support the new features or at most use existing instructions to work around,

thereby further increasing the program latency and the energy consumption.

To summarize, although the computing elements and the on-chip interconnects of today's DSRP can be made flexible and efficient with existing architectural and circuit techniques, the inflexible control architecture still limits the achievable efficiency and the hardware reusability.

4.2 Proposed 16-Core Universal DSP

The concept of *flexible* instruction-set architecture (ISA) is proposed to simplify the controls and, counter intuitively, enhance the flexibility of DSRPs. It starts with the observation that, for each particular task within the supported application domain, only a subclass of the entire instruction set is required. If we can flexibly define the necessary instructions prior to executing a task, then we no longer need a fixed and complex ISA to support all possible control patterns. Adaptation problems with design changes can also be resolved by simple hardware reconfigurations. To realize this concept, the traditional ISA decoder is replaced by several rows of register-based bit cells to store the task-dependent control signals. Then a simple selection logic is employed to access one row per clock cycle. Real-time redefinition is done by loading the bit cells with the contents of instruction memory through a specialized configuration command.

In addition to control circuits, we contribute with insights on design aspects regarding the core granularity and the connectivity for highly efficient and flexible implementations. The butterfly compute element (BCE) supports arbitrary 2×2 complex-valued

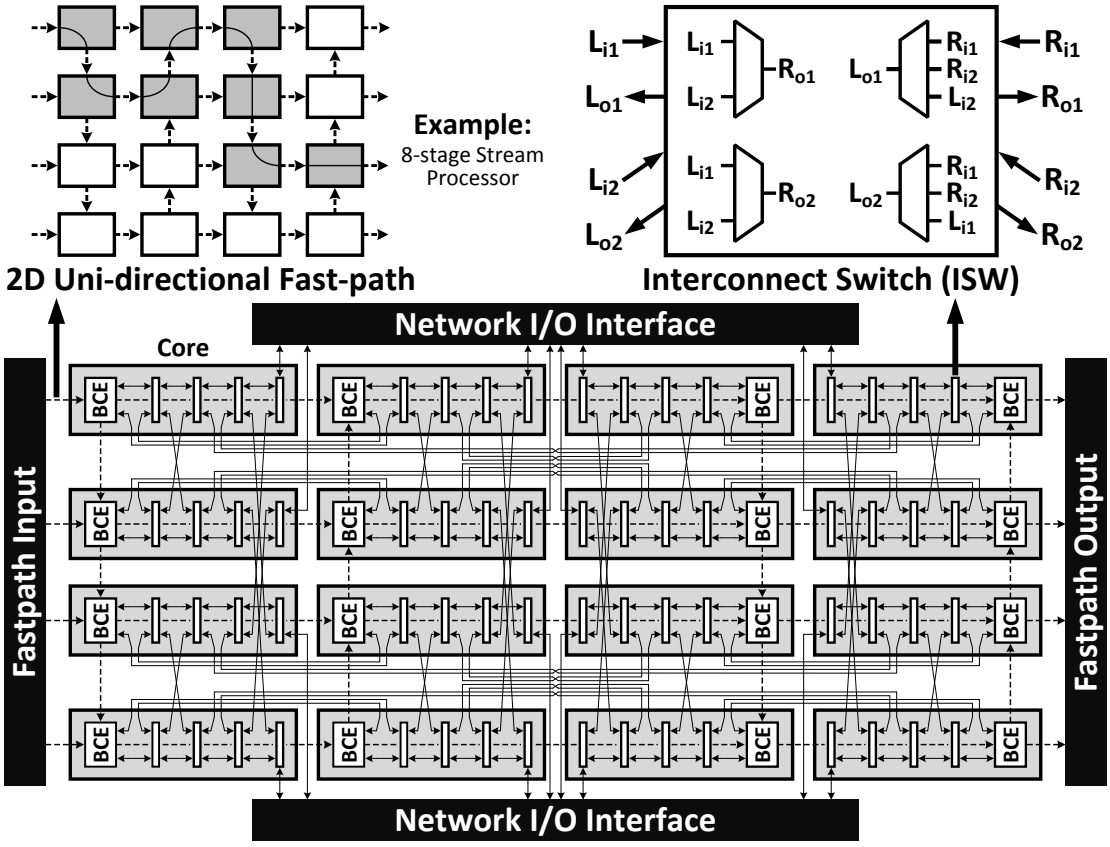


Figure 4.1: Processor architecture with multi-scale interconnects: fast-path (dashed lines) and radix-2 hierarchical network (solid lines).

matrix operations and is found as proper granularity for most of the baseband algorithms. The simple reason that drives the architectural decision is that, most of the SDR tasks are based on matrix operations, and the operations can be decomposed into multiple steps of 2×2 operations. The multi-level on-chip interconnects comprise the two-dimensional fastpaths and the radix-2 hierarchical butterfly network, allowing efficient information transfer between arrays of BCEs. Circuit techniques such as aggressive voltage scaling and fined-grain clock/input gating are applied for low power.

The proposed multi-core processor comprises 16 homogeneous cores in a 4×4 2D

array (Fig. 4.1). Each core embeds one BCE and four interconnect switches (ISWs) for programmable computing and networking. Local connection between adjacent cores is supported by a 128b uni-directional, 2D fast-path that allows horizontal transmission from left to right, and vertical link in a zigzag fashion. This approach saves 50% of the circuit-switched MUXes compared with a bi-directional scheme [65], yet it preserves enough connectivity for stream-oriented, multistage mappings. The ISWs form a radix-2 hierarchical network at the top level, allowing global data exchange and multicasting between cores and external interfaces.

4.2.1 Butterfly Compute Element

The datapath structure directly impacts energy efficiency of the core. An efficient dataflow minimizes memory accesses for data movement, reducing the program run-time and power. From our examination of common SDR algorithms (including those for multi-antenna (MIMO) applications), we propose a generic 2×2 butterfly dataflow structure as the proper granularity (Fig. 4.2). This 2×2 structure can directly map SDR functions such as FIR/IIR/Lattice filters, linear equalizer, CORDIC-based matrix decomposition, sphere decoder (SD), and others by simply concatenating multiple butterfly stages. The seemingly unrelated functions for spectrum shaping, channel factorization, and signal detection are compactly unified by the BCE.

The SIMD-style BCE is implemented to process 16b complex-valued data in fixed point (Fig. 4.3). Three major components, including 16b multimode multipliers, 32b shifters and 40b adders, are flexibly concatenated by the surrounding circuitry to per-

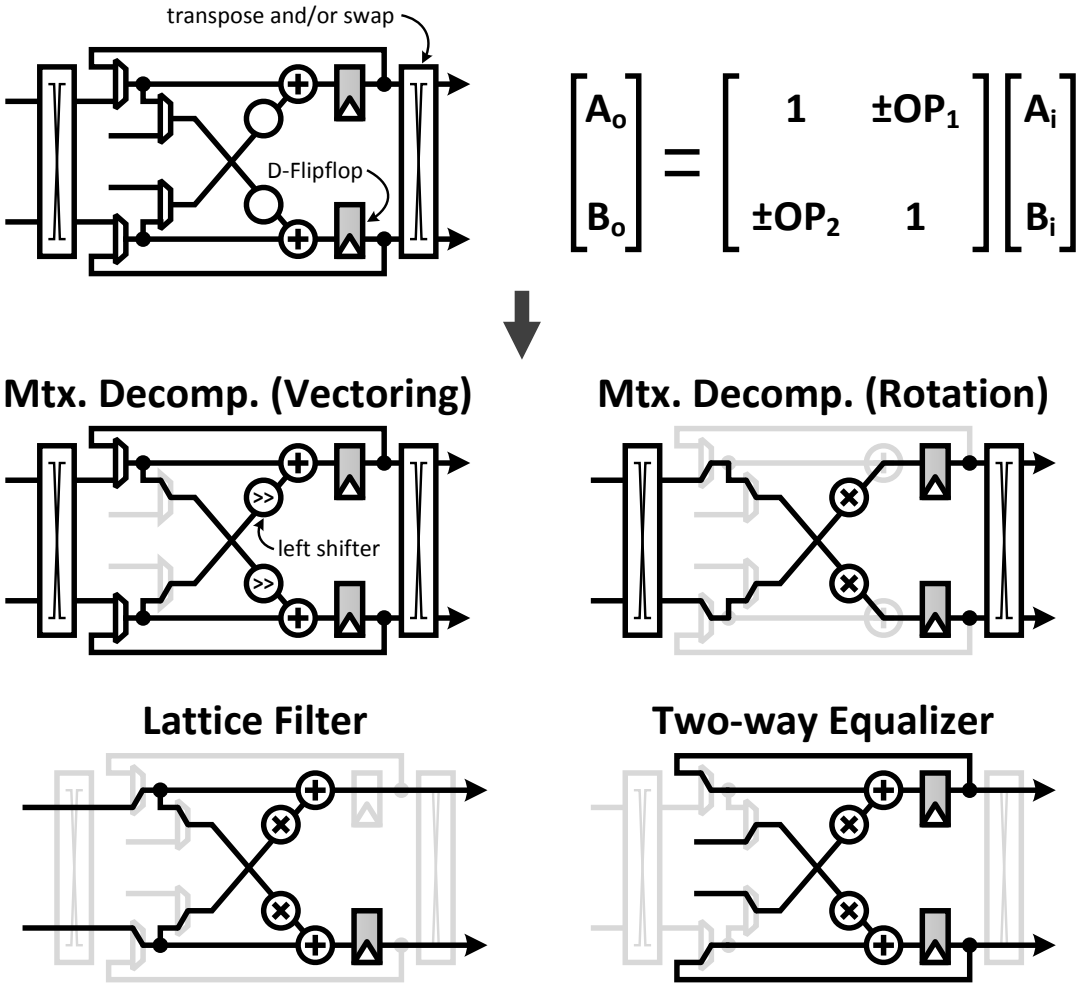


Figure 4.2: The generic 2×2 dataflow structure is considered as the proper granularity for SDR tasks.

form MAC, normalization, Euclidean distance, CORDIC, etc. Auxiliary components, such as CORDIC pre-scaling, sinusoid synthesis, and the metric-enumeration unit (MEU) accelerate the miscellaneous SDR computations. Two pipeline stages, plus the multiplier output registers, are inserted for a 500MHz clock rate, while the remaining registers are used for task-dependent retiming and data interleaving. All of the above components and memories are aggressively clock- or input-gated to save unnecessary switching energy. The multimode multiplier, as shown in Fig. 4.4, adopts the Baugh-Wooley and the three-dimensional partial-product reduction structure to support one 16b$\times$16b complex multiplication, two 16b$\times$16b real multiplications and one 4b$\times$16b complex subword-parallel MAC. Overall support for the multimode feature incurs 5.7% of BCE area overhead for 4$\times$ higher energy efficiency when performing successive interference cancellation (SIC), a critical block of multi-antenna signal detection. Specifically, the transmit/receive data symbols in SDRs can be represented by 4b complex value (4 bits for real and 4 bits for imaginary part), which is very inefficient if processed using 16b multipliers. Since the 16b$\times$16b multiplier can physically accommodate four 4b$\times$16b multipliers by simple datapath reconfiguration, the multi-mode approach can save up to 4$\times$ of multiplier resources in a four-antenna MIMO system.

The 2R2W 64$\times$32b register file (RF) is employed for local data access. When necessary, the RF can also be used to emulate the first-in-first-out (FIFO) buffer to adjust the processing latency. A diagram illustrating the local RF access is illustrated in Fig. 4.5. Following the 43b ISA example in Fig. 4.6, the access ports of the RF are fed by appropriate data based on the 3b header information and the 30b addressing

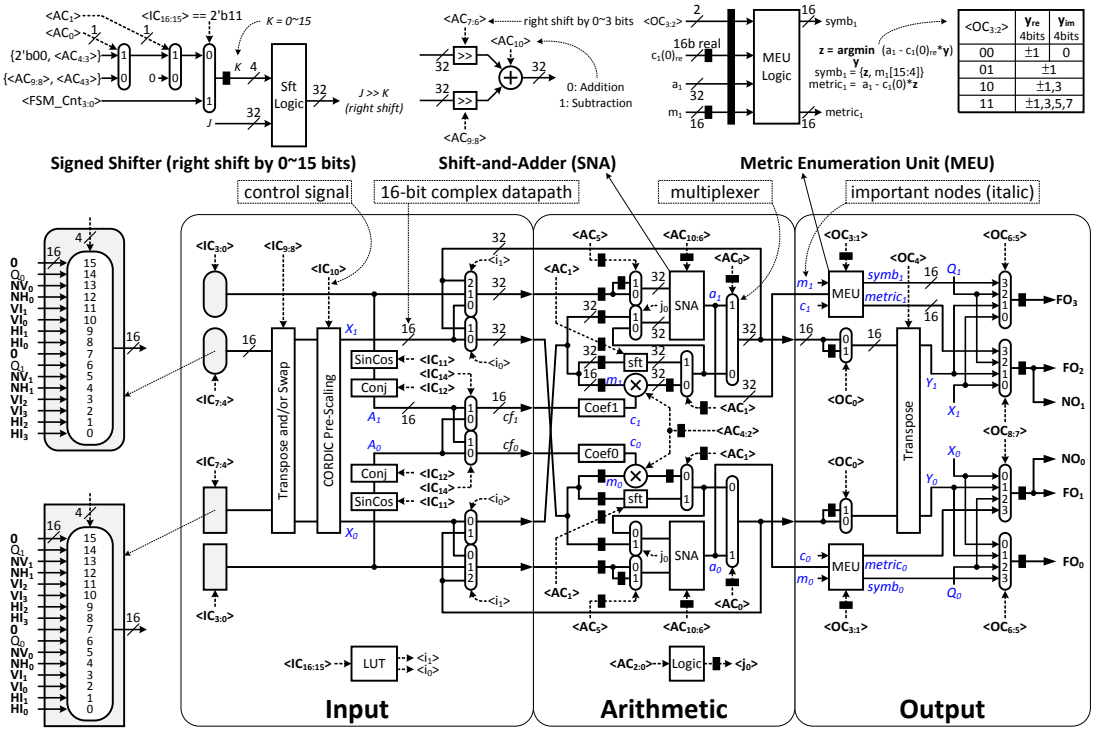
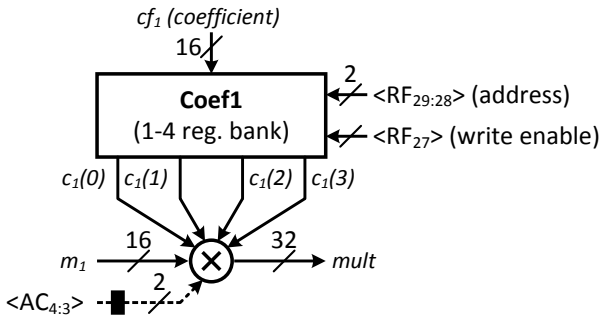


Figure 4.3: The detailed architecture of BCE.



<AC$_{4:3}$>	$mult = mult_{re} + jmult_{im}$
00	$m_{1re}^2 + jm_{1im}^2$
01	$m_{1re} * c_1(0)_{re} + jm_{1im} * c_1(0)_{im}$
10	$m_1 * c_1(0)$
11	$m_1[3:0] * c_1(0) + m_1[7:4] * c_1(1) + m_1[11:8] * c_1(2) + m_1[15:12] * c_1(3)$

Figure 4.4: The detailed architecture of the 16b multi-mode multiplier and the coefficient bank.

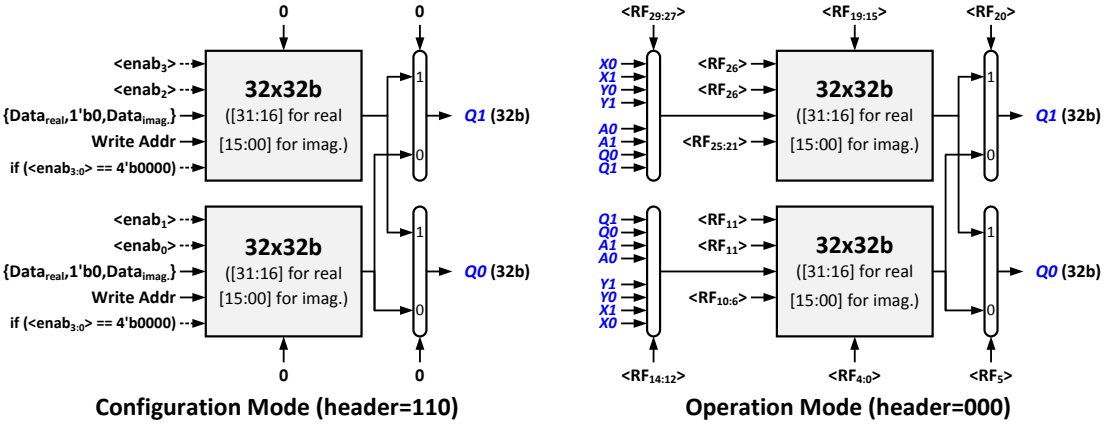


Figure 4.5: The detailed architecture of the 2R2W 64×32b register file.

control. The RF is uniquely composed by two independent sub-banks, M1 and M0, respectively. The write access can only be issued to different sub-banks, but the read access can come from the same sub-bank. This feature enables efficient data combining among M0 and M1 to lower the memory requirement for various SDR tasks, such as the candidate-symbol generation in the sphere decoding. The configuration mode also allows the instruction memory to directly write data into the RF, which is useful and efficient for the setup of algorithmic parameters.

We conclude this subsection by evaluating the efficiency of using this 2×2 coarse-grained compute element. Overall datapath consumes about 6μW/MHz per core at 1V supply, equivalent to an energy overhead from 50% to 210% as compared to the cost of active operations from heavy- to light-loaded tasks. This overhead, however, yields great reduction in memory access and program latency. A CORDIC square root takes 12 cycles with no intermediate RF access, which is 18× faster than 216 cycles in the state-of-the-art communication processor [67]. From an efficiency perspective, the

18× saving in instruction memory (4μW/MHz) access already outweighs the datapath overhead by 12×, without considering the overhead of RF and datapath in [67].

4.2.2 Flexible Instruction Set Architecture

Flexible ISA and state machine (SM) for task-specific control and run-time reconfiguration is proposed herein.

The nature of SDR workloads is exploited for energy-efficient and flexible BCE control. Although the complicated BCE datapath necessitates a large instruction set, the BCEs only use a small subset for each assigned task. A flexible ISA that can adjust itself to satisfy the task-specific needs, as a result, is more efficient than a complex hard-wired ISA with high instruction coverage but low utilization. The instruction access circuit of the flexible ISA can also be significantly simpler than which of the fixed ISA, since the flexible ISA only needs to define the necessary instructions for the targeted task instead of all possible instructions for every task.

The proposed control structure comprises a 128×43b instruction memory (IM), a programmable SM and a flexible ISA decoder (Fig. 4.6). In the 43b instructions, the 3b instruction header defines whether it belongs to the configuration or the operation type. The former configures the ISA decoder based on the remaining 40b content, while the latter uses a 10b opcode to access up to eight branching states, two network and 64 datapath configurations from the state, network and datapath register banks (SRB/NRB/DRB). The contents of the register banks are then selectively used to define the relationship between the opcode and the physical control pattern. Specifically, for

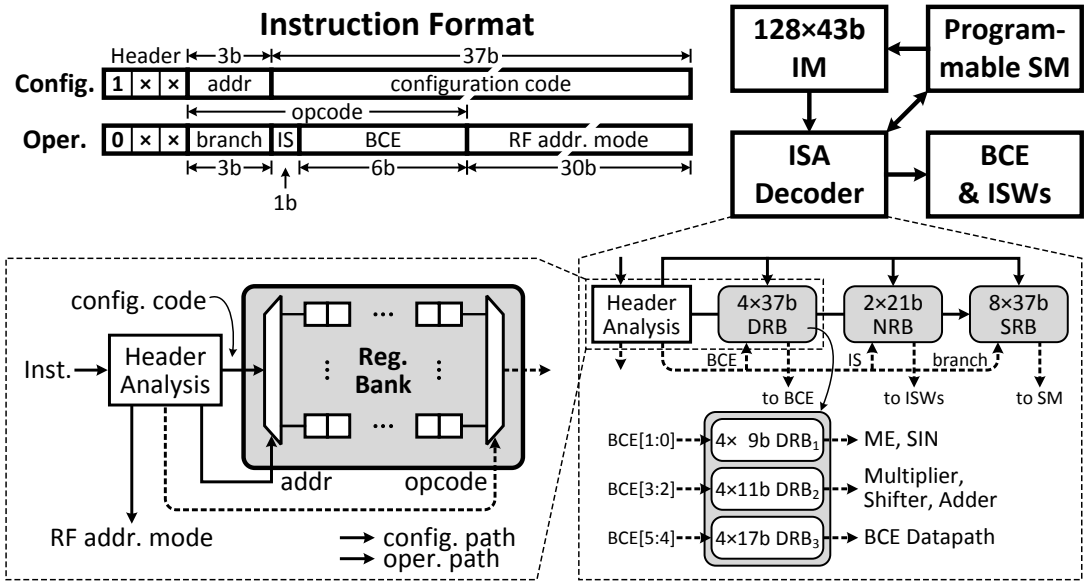


Figure 4.6: Flexible ISA control mechanism.

each task mapping, the BCE can configure the instruction decoder before it formally executes the operations. The configuration time is task-dependent, typically no more than 12 clock cycles. Due to the reconfiguration feature, the same opcode can result in totally different physical microcode information to distinct SDR tasks.

To save the computing complexity of ISA decoder, we partition the DRB into three sub-banks (DRB_{1-3}), each respectively controlling the input, the arithmetic, and the output part of the BCE. This partition, compared with a heuristic approach that requires a 64-row DRB to memorize 64 datapath configurations, can save the DRB area by 16× (4 vs. 64 rows) without significant loss in controllability.

The programmable SM directs the value and accumulation mode of the program counter (PC) to access contents of instruction memory. Each of the eight available states, as configured in the SRB, contains a 7-bit inner-loop bound, 7-bit outer-loop

bound, 1-bit PC accumulation enable, 7-bit PC start value, 1-bit state-counter accumulation enable, 5-bit state-counter start value, and a 5-bit state-counter end value to establish state transition and counting rules for a variety of pointers in the SM. The function of the pointers are explained as follows. When the 3-bit state pointer (sPtr) points to a new state, the 5-bit state counter is updated by the corresponding state register bank configuration to count from the state to the end value. The counter eventually turns on the state-transition flag (ST flag), which causes the sPtr to switch to the state specified by the 3-bit branch predicate. Meanwhile, the PC state value and the accumulation enable are switched to seamlessly deliver the IM address of the next subroutine. Three types of pointer banks, namely the inner-loop (iPtr), the outer-loop (oPtr) and the recovery pointers (rPtr) are utilized by the SM to store the temporary values of the inner loop, the outer loop, and the PC value of the subroutine calls in each state. When the ST flag toggles, the value of the iPtr of the current state increments by one, or it resets to zero if reaching the inner-loop bound (iloop flag turns on). The oPtr, meanwhile, increments when the iPtr resets, or resets to zero if it reaches the outer-loop bound (oloop flag turns on). The rPtr assists with subroutine retrievals when the same state is revisited. It memorizes the PC value plus one (PC+1) at the moment when the iloop flag is on, or resets to zero when the oloop flag is on. Overall, the SM allows for zero-overhead looping, branching and subroutine calls to map any SDR task with high efficiency and flexibility. Since the behavior of the SM is solely controlled by the state information in the SRB, the IM can fully focus on data manipulation without wasting time and energy on managing the PC value and the program flow. As a result, the SM

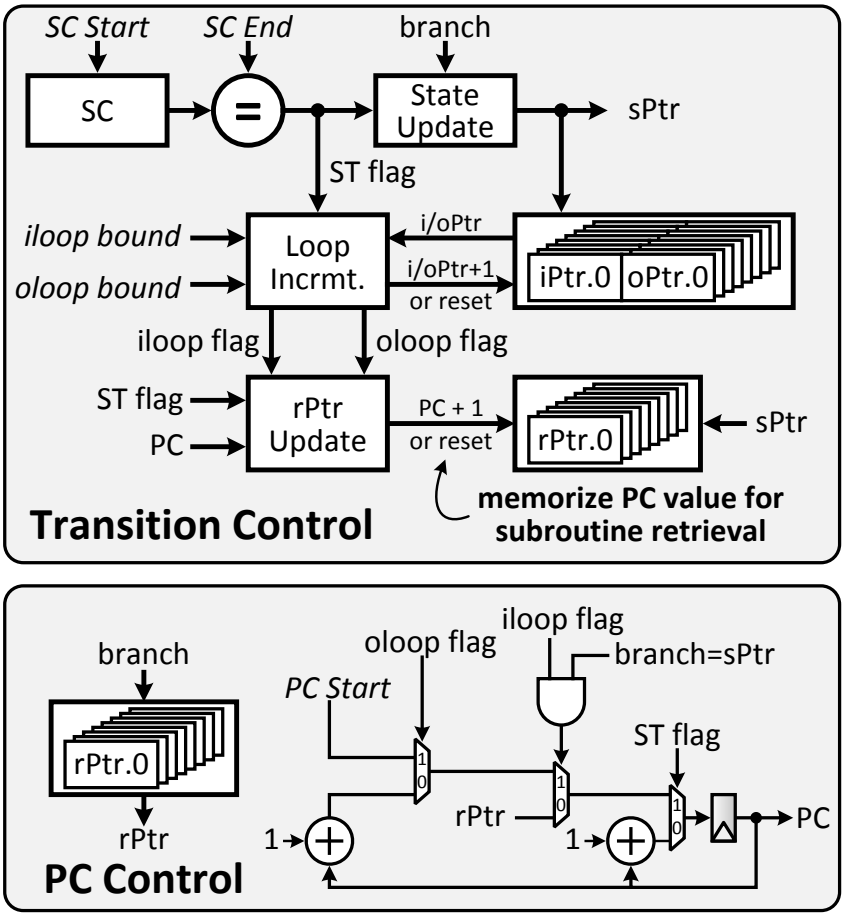


Figure 4.7: Chip integration flow and techniques for multi-scale interconnects.

can effectively lower the program runtime and emprove the IM efficiency as compared to traditional RISC/CISC processors.

The energy breakdown of the 4×4 QRD (memory- and control-dominant task) shows the benefits of flexible control (Fig. 4.8). Compared with the traditional control strategy, the flexible decoder dissipates 6.4× less energy due to simpler selection logic. The IM energy is also reduced by 3.3× due to the higher code efficiency and the SM. The overall energy saving is 1.8×, making the efficiency of QRD around

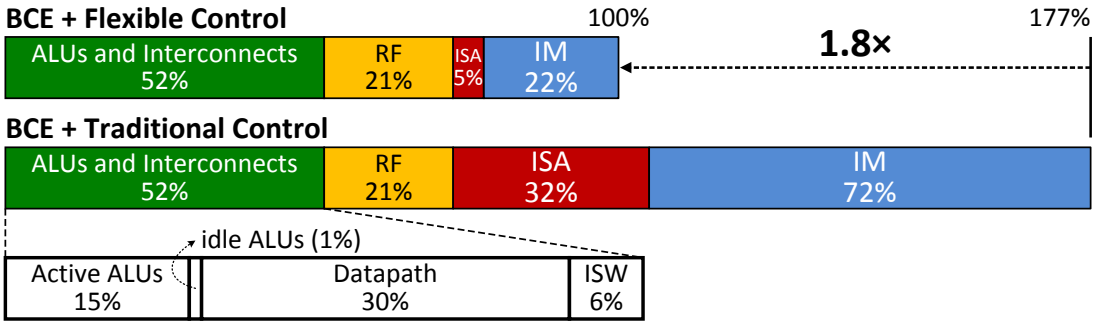


Figure 4.8: Energy breakdown of 4×4 QR decomposition shows an overall 1.8× of energy saving from the flexible-ISA over the traditional fixed-ISA scheme.

2.34GOPS/mW at 0.5V, only 2.3× lower than a dedicated QRD [70].

4.2.3 Interconnects and Top-Level Integration

The hierarchical network is adopted due to its better scalability than the 2D mesh. It connects the 64b network I/O from each BCE, offering full connectivity and up to 512Gbps bisection bandwidth (80Gbps per core with fast-path I/O) at 500MHz. Integrating the cores with fast-path and hierarchical network in a dense layout can easily incur performance loss due to critical-wire delay. We tackle this challenge by selectively reserving lower metal layers for feed-through routing and buffer insertion, and optimizing the timing across the core boundaries using interface logic modeling (Fig. 4.9). This approach enables a 10× improvement on wire delay and 30% more routing resources for cores, compared with a heuristic bottom-up method. A 95% silicon utilization is achieved for a total of 10M transistors, including level shifters inserted at I/O interfaces for aggressive voltage scaling down to 415mV.

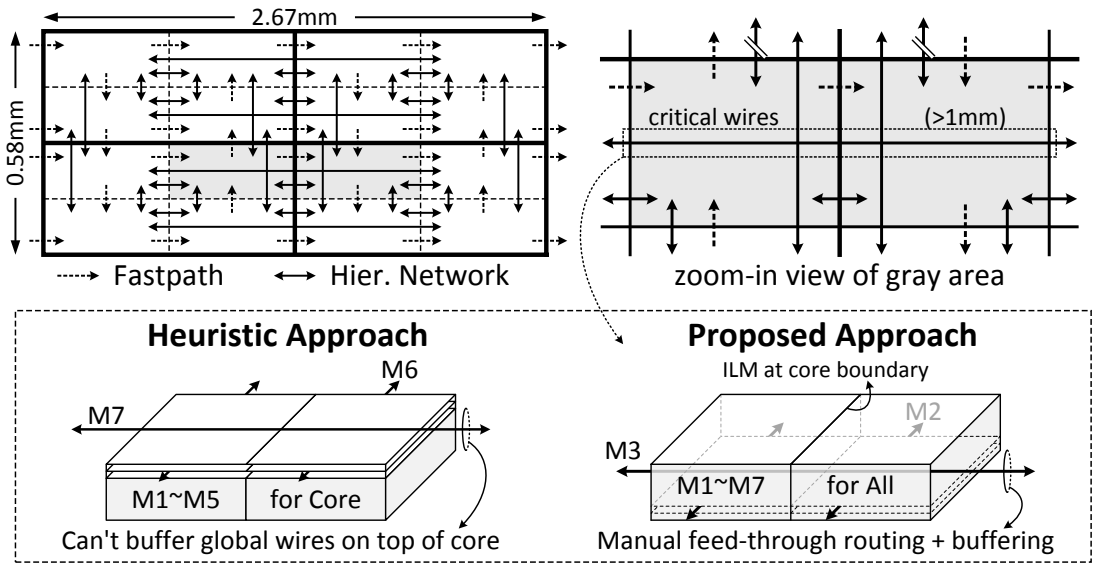


Figure 4.9: Chip integration flow and techniques for multi-scale interconnects.

4.3 Programming Model

We profile the data- and control-flow mapping of each task and develop a custom assembler to efficiently define the ISA, allocate the BCE resources, and route the interconnects as exemplified in Fig. 4.10 for an IIR. Specifically, the dataflow structure of each task can be modeled as a "template," so that whenever we want to map a particular algorithm, we can simply call the corresponding template that already optimizes the datapath, the state transition rule and the instructions in each BCE. The only thing remains undefined is the interconnect between BCEs, and that has to be decided based on the locations of available BCE for optimal utilization of local and hierarchical interconnect fabrics. Typically for stream-oriented, task-level-parallel mappings we will use as much local fastpath as possible, while for designs with feedback paths (e.g. IIR filtering) we will also incorporate hierarchical network to assist the routing.

A simple custom assembler, made with Windows Excel, accepts human-readable commands and transforms them into binary machine codes. The codes are then fed to the test chip using a 800-bit scan-chain (50 bits for each BCE, where 43 bits are for instructions, and 7 bits are the corresponding addresses in the IM). If the BCE requires re-programming during chip operations, it will need to call a special REQ instruction and point its hierarchical network directly to the I/O interface[1]. Once the external interface gets the request, it will send the new instructions to the BCE via hierarchical network, so the BCE's IM can be refreshed. Refreshing multiple IMs is made possible by also configuring each BCE with a "core ID," so the new instructions for a particular ID will be only accepted by those BCEs whose ID matches. This broadcasting approach to refresh the IM(s) can enable seamless architectural transformation and allow interaction between local IM and high-level caches.

4.4 Measurements and Comparisons

The design is integrated in a 2.67mm$\times$0.58mm tile of a 24.5mm^2 in-house FPGA as an SDR acceleration processor (Fig. 4.11). Five representative benchmarks are mapped for standalone verification: (1) 64-tap raised-cosine FIR [68]; (2) 8th-order Chebyshev Type-II IIR; (3) 4th-order cyclic-autocorrelation (CAC) [71]; (4) CORDIC-based 4$\times$4 matrix decomposition [70]; (5) 4$\times$4 MIMO SD [69]. Figure 4.12 illustrates some mapping details and the energy efficiency vs. supply voltage. A 5-5.6$\times$ efficiency gap is observed between the compute-centric SD and the memory-/control-dominant

[1]The REQ command is specially designed to only transmit via the hierarchical network.

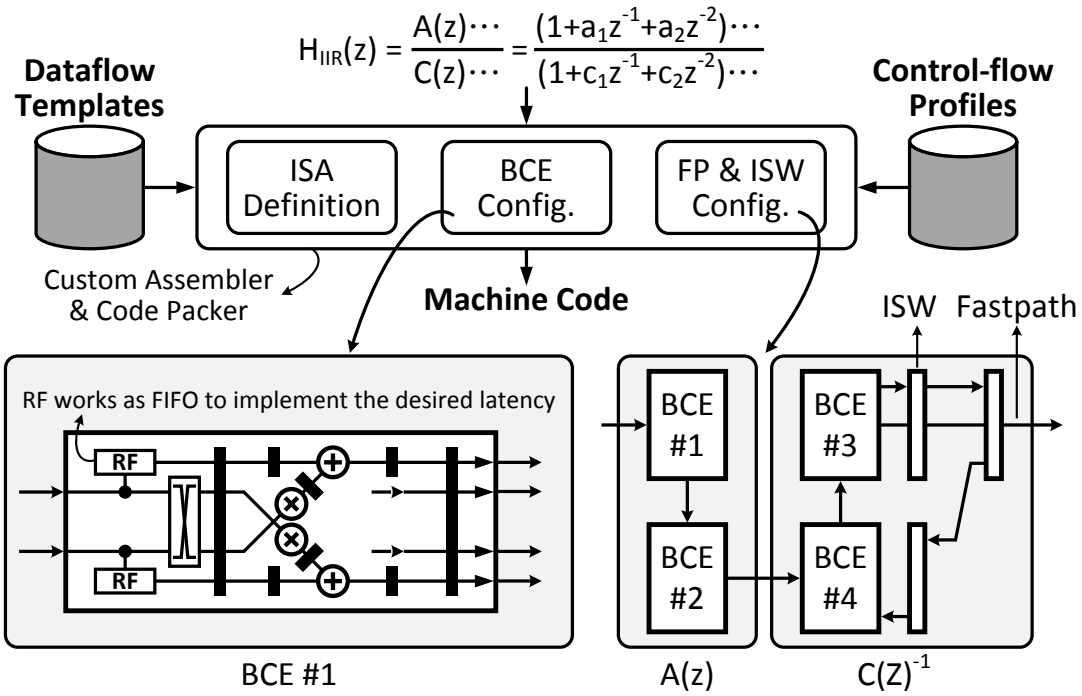
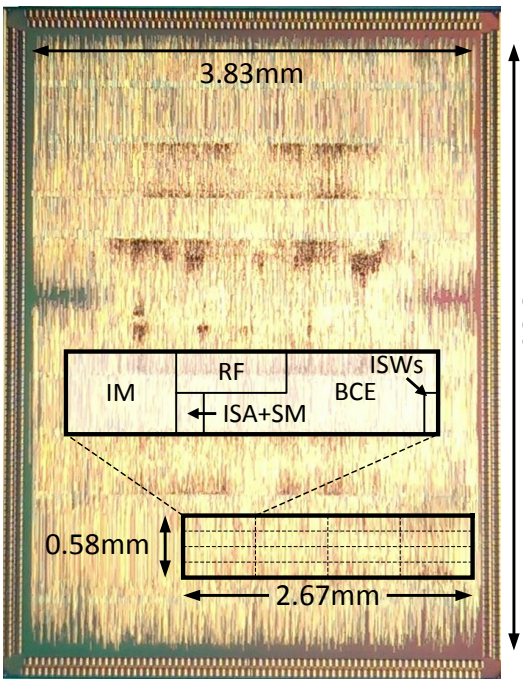


Figure 4.10: Chip programming model and mapping example.



Technology	TSMC 40nm CMOS
Area	1.5mm^2 (10M Transistors)
Voltage	0.42V - 1.0V
Frequency	25MHz - 500MHz
Average Power	0.28mW - 17.18mW (per core)
Nom. Energy Efficiency	5.2GOPS/mW (192fJ/OP) @0.9V, 400MHz
Peak Energy Efficiency	13.1GOPS/mW (76fJ/OP) @0.42V, 25MHz

Figure 4.11: Chip micrograph and performance summary.

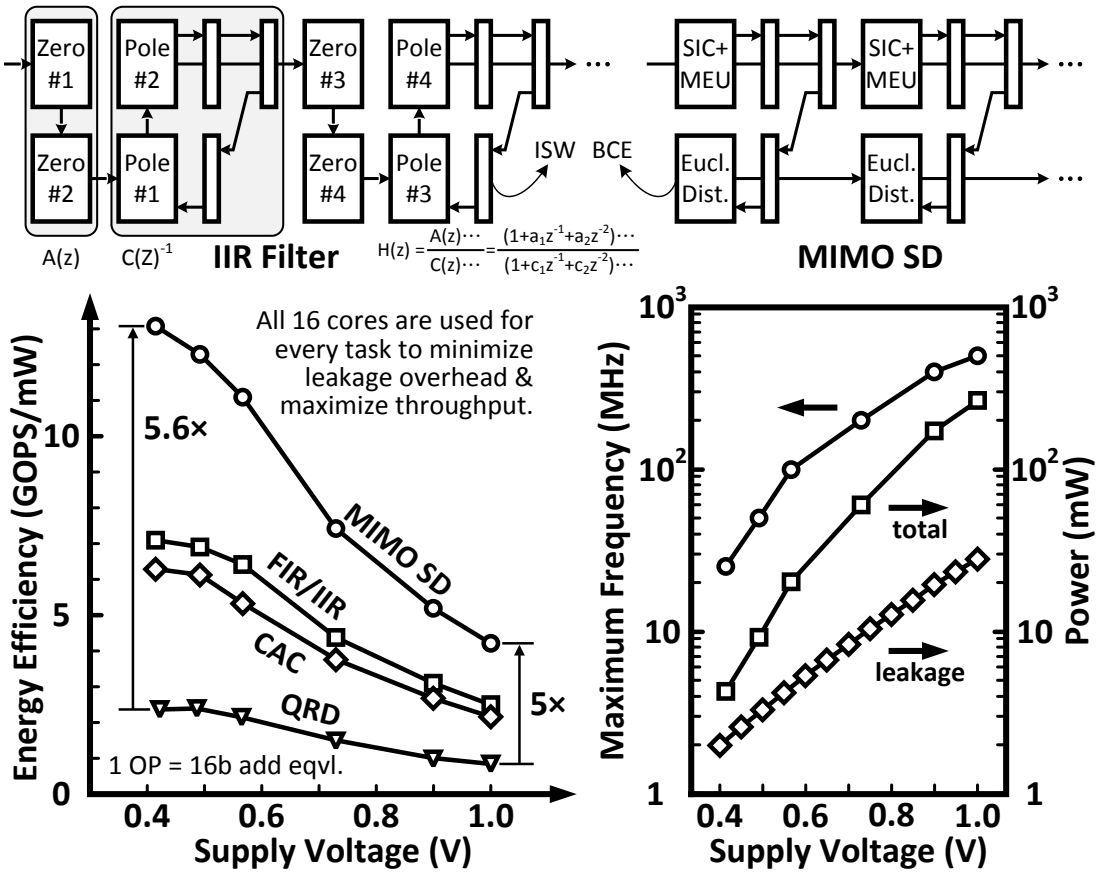
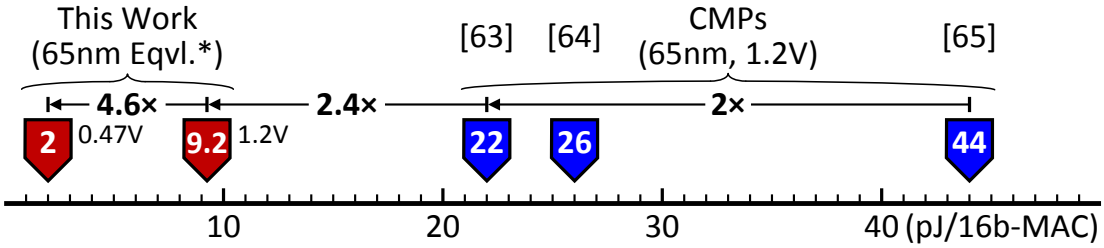


Figure 4.12: Benchmark mapping examples and performance measurements in 40nm CMOS.

QRD. Robust functionality is measured down to 415mV with minimum power of 275 μW/core at 25MHz, 25-degree Celsius for a 4×4 MIMO SD. At this operating point, the leakage and the active power are roughly equal. This point also translates to a peak energy efficiency of 13.1GOPS/mW (76fJ/OP). The performance of the chip can scale up to 1.17TOPS at 500MHz, 1.0V, achieving 4.2GOPS/mW.

To validate the efficiency benefits over state-of-the-art communication processors, we normalize the energy per operation of this work to 65nm for a fair comparison. The



Benchmarks	ASIC Designs	This Work*		Effi. Gap
CAC (300MS/s)	4.3mW, 0.5V ([71], 40nm)	5.8mW, 0.44V		1.35×
FIR Filter (190MS/s)	130mW, 1.1V ([68], 90nm)	40nm	50.1mW, 0.73V	1.59×
		90nm Eqv.	207mW, 1.1V	
4×4 MIMO SD (323Mbps)	40mW, 0.9V ([69], 130nm)	40nm	16.8mW, 0.51V	2.62×
		130nm Eqv.	105mW, 0.9V	
4×4 QR Decmp. (3.48MConv./s)	25mW, 0.9V ([70], 130nm)	40nm	8.8mW, 0.49V	2.34×
		130nm Eqv.	58.4mW, 0.9V	

*Normalization is performed by matching the gate delay vs. supply voltage, and scaling the leakage and active power separately across technology nodes based on SPICE simulations.

Figure 4.13: Comparisons with state-of-the-art communication multiprocessors and functionally-equivalent ASICs.

normalization is performed by matching the gate delay vs. supply voltage, and scaling the leakage and active power separately across technology nodes based on SPICE simulations. A normalized energy of 9.2pJ/16b-MAC shows 2.4-4.8× higher efficiency than [63]-[65] (Fig. 4.13). The improvement scales up to 13× for low-voltage operations since [63] [64] lack voltage scalability. Since energy per operation alone doesnt guarantee real-time performance, we normalize the energy of this work and functionally-equivalent ASICs [68]-[71] to the same throughput for a fair comparison (Fig. 4.13). Our design bridges the energy efficiency gap with ASICs to within 2.6×.

4.5 Summary

A 16-core processor for software-defined radios is realized in 40nm CMOS. Key design techniques are the algorithm-architecture co-design to determine the proper core granularity, and the flexible instruction-set architecture that greatly reduce the control and energy overhead, while providing the freedom to adapt to design changes. Integration with the long-wire interconnects has to be carefully handle to prevent performance loss due to critical-wire delay. The solution to the interconnect problem is to manually reserving low-layer metals for feed-through routing and buffer insertion.

The processor features domain-specific kernels, flexible control and multi-scale interconnects. It achieves a peak energy efficiency of 13.1GOPS/mW (76fJ/OP) at 415mV, 25MHz, and a peak performance of 1.17TOPS at 1V, 500MHz, showing $>2.4\times$ higher energy efficiency than state-of-the-art communication chip multiprocessors, and closing the gap with functionally-equivalent ASICs to within $2.6\times$.

CHAPTER 5

Chip Verification Methodology

This chapter presents the chip verification procedure, including the in-house assembler, the printed-circuit board design, and the FPGA-based pattern generation and data analysis.

5.1 Generating Machine Code for Programmable Chips

Since the design examples demonstrated in this dissertation are programmable processors, the evaluation process will be more difficult than the cases for typical ASICs. To efficiently generate the machine code, we develop an in-house assembler using Windows Excel software (Fig. 5.1). The assembler accepts human-readable commands and transform them into binary machine codes. The codes are then stored in the FPGA's block RAM (BRAM) and fed to the test chip using a scan-chain. The scan-chain includes the information of the instructions and the corresponding memory addresses. The chain firstly shifts in the information of a instruction and its address, and then it halts and waits for an external write-enable (WE) signal to toggle high to configure the memory. When WE resets to zero, the scanning continues to send the next set of com-

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA
5.CnfgFSM	100	FSM.0	9	0	c.ACCUM	0	c.ACCUM	0	0	b.ByMEM	b.ByMEM	b.ByMEM	b.ByMEM	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx
5.CnfgFSM	100	FSM.1	1	0	b.IDLE	9	b.IDLE	0	0	b.ByMEM	b.ByMEM	b.ByMEM	b.ByMEM	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx
5.CnfgFSM	100	FSM.2	1	0	b.IDLE	9	b.IDLE	0	0	b.ByMEM	b.ByMEM	b.ByMEM	b.ByMEM	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx
5.CnfgFSM	100	FSM.3	1	0	b.IDLE	9	b.IDLE	0	0	b.ByMEM	b.ByMEM	b.ByMEM	b.ByMEM	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx
6.CnfgALU	101	b.ALU.0	2.Y	2.Y	1.OFF	1.BPSK	1.OFF	1.OFF	1.ADD	SNA>>0	SNA>>0	3.CPLX	1.OFF	2.ON	1.OFF	1.OFF	1.LATTICE	2.A1	1.A0	1.A+jB	1.OFF	1.OFF	2.ON	1.OFF	d.VI.30	d.VI.30
6.CnfgALU	101	c.ALU.1	0.OSEL_21	0.OSEL_30	0.Trans	0.MEU_Mc	0.MEU_En	0.DFF_En	0.SNA_Md	SNA_21	SNA_30	0.MUL_Mc	0.DFF_En	0.MUL_En	0.SFT_En	0.SNA_En	0.DataCtrl	0.Coef_Up	0.Coef_Dn	0.Conj	0.Sin_En	0.CDC_En	0.Swap	0.Trans	a.ISEL_21	a.ISEL_30
6.CnfgALU	101	d.ALU.2	0.OSEL_21	0.OSEL_30	0.Trans	0.MEU_Mc	0.MEU_En	0.DFF_En	0.SNA_Md	SNA_21	SNA_30	0.MUL_Mc	0.DFF_En	0.MUL_En	0.SFT_En	0.SNA_En	0.DataCtrl	0.Coef_Up	0.Coef_Dn	0.Conj	0.Sin_En	0.CDC_En	0.Swap	0.Trans	a.ISEL_21	a.ISEL_30
6.CnfgALU	101	e.ALU.3	0.OSEL_21	0.OSEL_30	0.Trans	0.MEU_Mc	0.MEU_En	0.DFF_En	0.SNA_Md	SNA_21	SNA_30	0.MUL_Mc	0.DFF_En	0.MUL_En	0.SFT_En	0.SNA_En	0.DataCtrl	0.Coef_Up	0.Coef_Dn	0.Conj	0.Sin_En	0.CDC_En	0.Swap	0.Trans	a.ISEL_21	a.ISEL_30
7.CnfgNWK	110	Zero*5	Zero*5	Zero*5	00	b.RIO	b.RIO	b.LIO	b.RIO	b.RIO	b.LIO	b.LIO	b.RIO	b.RIO	b.LIO	b.LIO	b.RIO	b.RIO	b.LIO	b.LIO	xx	xx	xx	xx	xx	xx
2.CnfgCoef	001	FSM.1	2.ON	OPT.0	ALU.0	IPT.0	2.ON	0	1.OFF	0	0	2.ON	0	1.OFF	0	0	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx
0.CMD	000																									
5.CnfgFSM	100	FSM.0	13	0	c.ACCUM	0	c.ACCUM	0	0	b.ByMEM	b.ByMEM	b.ByMEM	b.ByMEM	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx
5.CnfgFSM	100	FSM.1	0	0	b.IDLE	13	b.IDLE	5	0	b.ByMEM	b.ByMEM	b.ByMEM	b.ByMEM	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx
5.CnfgFSM	100	FSM.2	13	0	c.ACCUM	20	b.IDLE	1	71	b.ByMEM	b.ByMEM	b.ByMEM	b.ByMEM	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx
5.CnfgFSM	100	FSM.3	9	1	c.ACCUM	22	b.IDLE	0	0	b.ByMEM	b.ByMEM	b.ByMEM	b.ByMEM	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx
5.CnfgFSM	100	FSM.4	1	0	c.ACCUM	24	c.ACCUM	0	0	b.ByMEM	b.ByMEM	b.ByMEM	b.ByMEM	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx
6.CnfgALU	101	b.ALU.0	3.M	3.M	1.OFF	0.MEU_Mc	1.OFF	1.OFF	1.ADD	SNA>>3	SNA>>0	3.CPLX	1.OFF	2.ON	1.OFF	1.OFF	2.MAC	2.A1	1.A0	2.A-jB	2.ON	1.OFF	2.ON	1.OFF	f.NH.10	g.NV.10
6.CnfgALU	101	c.ALU.1	0.OSEL_21	0.OSEL_30	0.Trans	0.MEU_Mc	0.MEU_En	0.DFF_En	1.ADD	SNA>>3	SNA>>0	3.CPLX	1.OFF	2.ON	1.OFF	2.ON	0.DataCtrl	0.Coef_Up	0.Coef_Dn	0.Conj	0.Sin_En	0.CDC_En	0.Swap	0.Trans	a.ISEL_21	a.ISEL_30
6.CnfgALU	101	d.ALU.2	3.M	3.M	2.ON	0.MEU_Mc	1.OFF	2.ON	1.ADD	SNA>>0	SNA>>0	0.MUL_Mc	1.OFF	1.OFF	2.ON	2.ON	1.LATTICE	1.A0	1.A0	1.A+jB	1.OFF	2.ON	1.OFF	2.ON	h.M.10	q.Zero.1
6.CnfgALU	101	e.ALU.3	0.OSEL_21	0.OSEL_30	0.Trans	0.MEU_Mc	0.MEU_En	0.DFF_En	0.SNA_Md	SNA_21	SNA_30	0.MUL_Mc	0.DFF_En	0.MUL_En	0.SFT_En	0.SNA_En	0.DataCtrl	0.Coef_Up	0.Coef_Dn	0.Conj	0.Sin_En	0.CDC_En	0.Swap	0.Trans	a.ISEL_21	a.ISEL_30
7.CnfgNWK	110	Zero*5	Zero*5	Zero*5	00	b.RIO	b.RIO	b.LIO	b.RIO	b.RIO	b.LIO	b.LIO	b.RIO	b.RIO	b.LIO	b.LIO	b.RIO	b.RIO	b.LIO	b.LIO	xx	xx	xx	xx	xx	xx
7.CnfgNWK	110	Zero*5	Zero*5	Zero*5	00	b.RIO	b.RIO	b.LIO	b.RIO	b.RIO	b.LIO	b.LIO	b.RIO	b.RIO	b.LIO	b.LIO	b.RIO	b.RIO	b.LIO	b.LIO	xx	xx	xx	xx	xx	xx
7.CnfgNWK	110	Zero*5	Zero*5	Zero*5	00	b.RIO	b.RIO	b.LIO	b.RIO	b.RIO	b.LIO	b.LIO	b.RIO	b.RIO	b.LIO	b.LIO	b.RIO	b.RIO	b.LIO	b.LIO	xx	xx	xx	xx	xx	xx

Figure 5.1: Custom assembler development using Windows Excel software.

mand and address until all required information is properly accepted by the processor.

5.1.1 Printed Circuit Board

The Altium Designer tool by Altium Ltd. is employed for the printed-circuit board (PCB) designs. Proper footprints of the discrete components (e.g. package, voltage regulator, and capacitance) are drawn by the user to establish a design library for global schematics, placement and routing. Figure 5.2 exemplifies the PCB design for the classification processor. The board uses a 4-layer FR4 material with overall thickness of 0.062 inches. Three voltage regulators provide stable voltage translation from the global 3.3V supply, delivering 1.8V for chip I/O cells, 0.9V for chip memories and scalable 0.35-0.90V for low-power logics (e.g. MAA kernels in the FEX engine, as shown in Section 3). One of the new contributions this PCB achieves is the use of FPGA mezzanine card (FMC) connectors [77] [78] [79]. The 400-pin FMC connector offers high-speed data transceiving and has reserved a lot of pins for special functionalities

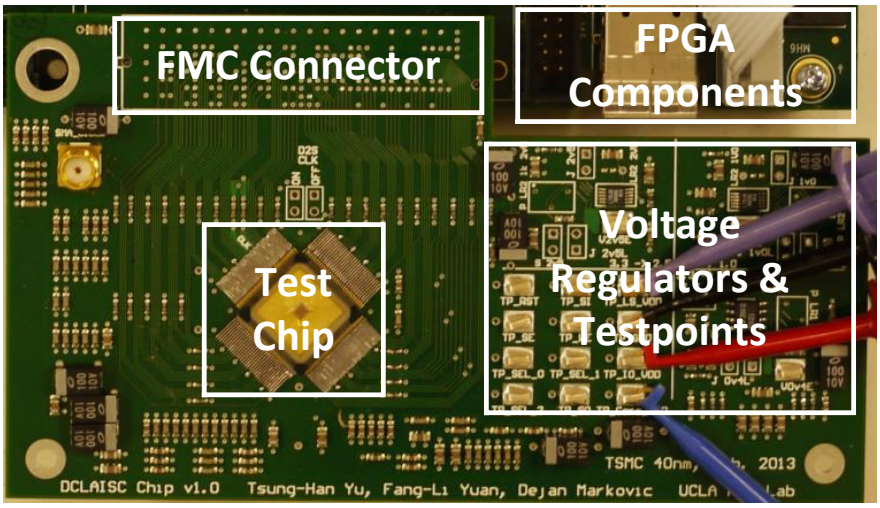


Figure 5.2: PCB design for the classification processor. The voltage regulators and the FMC connector are placed on the right and top, respectively. The PCB is made in L shape to avoid touching the FPGA components.

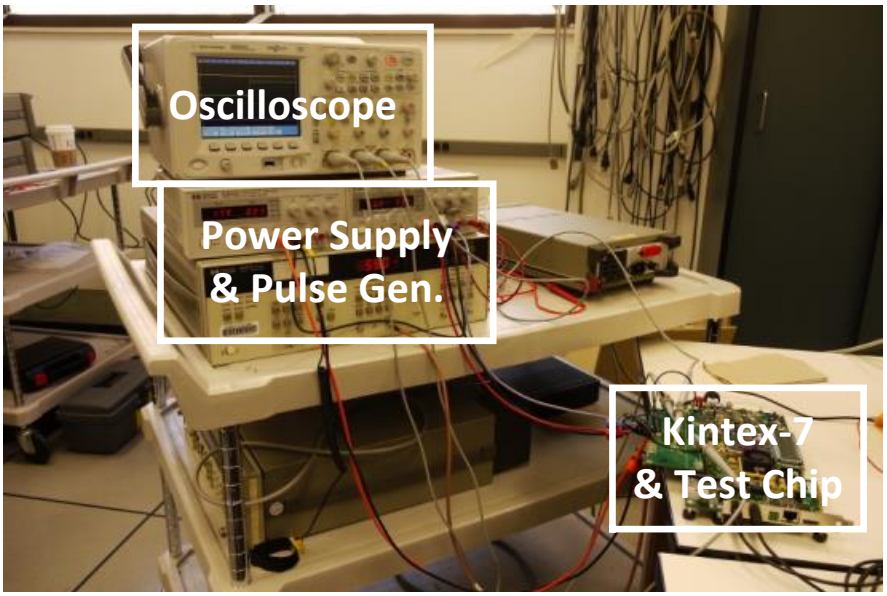
such as JTAG interfaces. The design of a PCB with FMC connectors, however, requires special attention on the board dimension, shape and the pin setup. For the case of Kintex-7 FPGA board we use for chip testing, since there are some I/O slots right next to the FMC connector on the FPGA, we have to make the PCB "L-shape" to avoid touching and blocking those FPGA components. The other thing worths attention is that the FMC connectors have three special pins as the input, output, and enable flag of the JTAG chain from the FPGA. In normal cases, the FPGA routes the JTAG chain internally to program its components. However, if the enable flag is connected to ground, the FPGA will notice the existence of a new board connecting through the FMC connector, thus it will route the JTAG signal through the FMC and try to program that board altogether. For chip testing in this dissertation, since we use a separate scan-

chain to program the designs, we don't rely on JTAG interface. As a result, the enable flag pin *must be kept floating* to pretend there is no board connected to the FPGA. In the case of our chip testing, that particular pin is at the H2 location of the FMC connector.

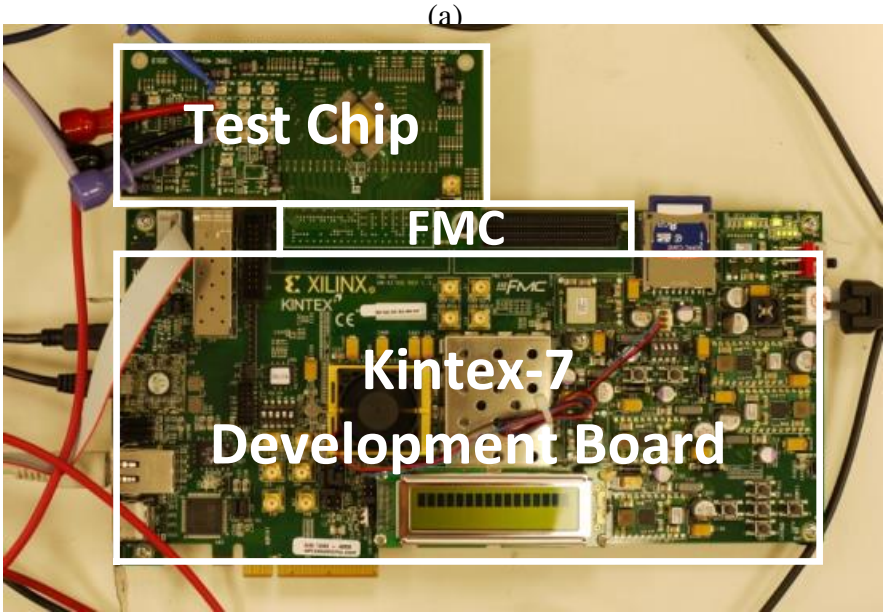
5.1.2 FPGA-based Patter Generation and Data Analysis

We use Xilinx Kintex-7 development FPGA for ASIC verification [80]. Abundant on-board resources (500 multi-standard I/Os, 16Mb Block RAM, 840 DSP48 and 5.1k Logic Slices) enable highly complex pattern generation and data analysis. External memory modules (1GB DDR3 SODIMM at 1.6Gbps, 16MB Quad SPI Flash, SD Card Slot and 128MB Linear BPI Flash for PCIe Configuration) further support large-size data accessibility. Equipped with two high-speed, 400-pin FMC connectors, the Kintex-7 board can be expanded to communicate with commercial RF modules, debug cards, and ASIC designs. A wide range of operating frequency from 10 to 910MHz, either generated on board or externally, is tunable during functional verification to obtain accurate relations between power, voltage and clock rate. Integrated software environment (ISE) and JTAG/I^2C-based configuration enable user-friendly, efficient bit-file generation and FPGA programming.

As a demonstration, Fig. 5.3 shows the measurement setup of a blind-signal classification IC in 40-nm CMOS for cognitive radios. The measurement setup includes the Kintex-7 FPGA, the Agilent MSO6104A oscilloscope, the power supply, and the (optional) pulse generator. The FPGA serves as a testbench and send the data over to the test chip at up to 500Mb/s per FMC pin. The output of the chip is sent back via the



(a)



(b)

Figure 5.3: Chip measurement setup with (a) external equipments and Xilinx Kintex-7 FPGA board. Detailed setting between the FPGA board and the test chip in shown in (b).

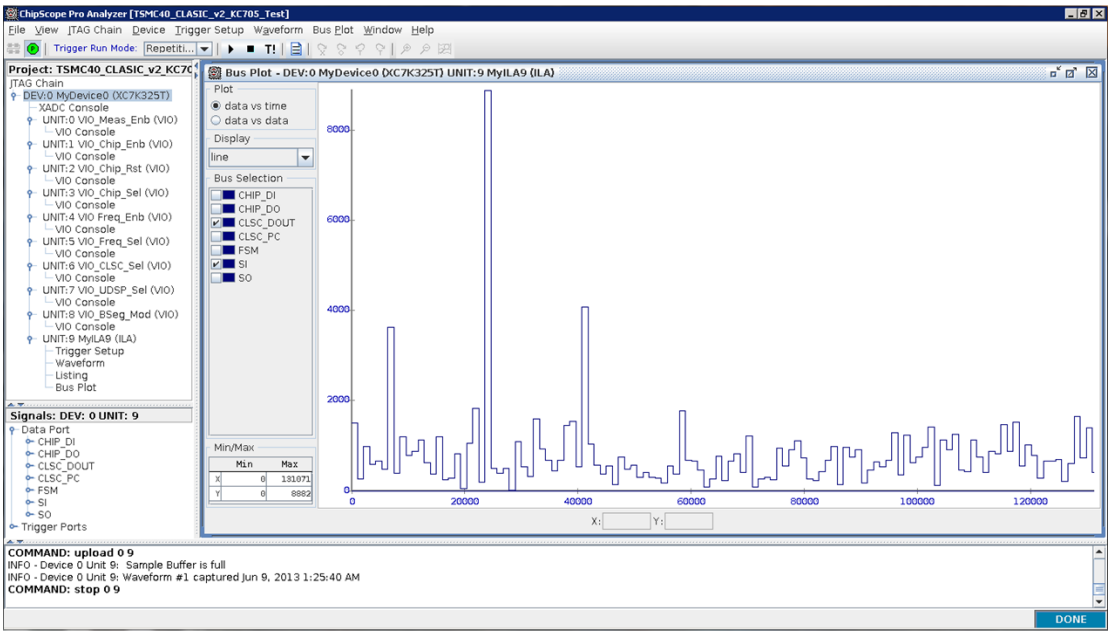


Figure 5.4: Real-time verification using Xilinx ChipScope software. The measured data can be arranged to the proper binary, decimal or hexadecimal formats for better readability. More advanced features are detailed in the user guide [81].

same interface and plotted using the Xilinx ChipScope software [81] for real-time verification (Fig. 5.4). One thing worth noticing is the I/O standards. The Kintex-7 board provides I/O standards (e.g. SSTL, HSTL adn LVDS) with seven different selectable voltages for compatible data transceiving [82]. The I/O standards can be decided in ISE software during FPGA configurations, but the voltages require a third-party IC from Texas Instruments. to adjust. As a result, an additional USB adaptor and the Fusion Digital Power Designer software [83] are required to change the voltage of the FPGA's I/O banks. For the case of our designs, we need 1.8V supply to safely handle the 1.8V I/O cells at the chip boundary.

5.2 Summary

The verification of programmable processors requires an efficient software to generate the machine code with as least amount of human work as possible. Entrusting the computer to assist code generation can also minimize the error from manual compilation. The printed circuit board that embeds the high-speed FMC connector requires special attention on the board shape/dimension and the pin definition. Carefully reading documents of FMC usage is strongly requested. The Xilinx Kintex-7 FPGA embeds abundant on-chip programmable fabrics, memories and I/O transceivers for real-time pattern generation. A third-party USB adaptor from Texas Instruments, however, is required to property setup the I/O voltage for compatible data transceiving between the test chip and the FPGA. Data analysis can be easily realized by capturing the output of the test chip through the FMC connector, and plotted using Xilinx ChipScope. The user-friendly graphic interface speeds up the verification and easily interests the audience during real-time demonstration.

CHAPTER 6

Conclusion

This dissertation pursuits architectures that enable the coexistence of flexibility and efficiency in one chip, and uses the next-generation software-defined and cognitive radios as a demonstration platform. Following the method of *hybridizing* the design concepts taken from the dedicated hardware and the programmable processor, four major ideas are proposed to fulfill the research goal. These ideas are (1) Algorithm-architecture co-design by analyzing the workload distribution and the tradeoff between dependent processing blocks, and performing spatial mapping to determine the core granularity; (2) Dynamic parallelism-frequency scaling (DPFS) and multi-core power management to achieve near-optimal energy efficiency regardless of the throughput requirements; (3) Flexible instruction-set architecture (ISA); (4) Multi-scale interconnects to define the task-dependent control patterns at system runtime instead of chip design time, thereby enabling the freedom to define new instructions, more efficient usage of instruction memory, and potentially shorter program runtime with the help of programmable state machine.

These concepts are applied to design the blind signal classifier for cognitive radios (CRs), and the multi-core baseband DSP for software-defined radios (SDRs), using

40nm CMOS technology. The classification DSP features three-step parameter estimation for a 59× energy saving compared to an exhaustive method, and multi-algorithm feature extraction to distinguish five modulation classes: multicarrier, single-carrier PSK/QAM/MSK, and spread-spectrum signals. The chip consumes 17μJ within 2ms sensing time per classification, achieving 95% detection probability and 0.5% false-alarm rate at 10dB SNR in a 500MHz channel. Classification at 0dB SNR requires around 15× higher energy due to longer processing time, but the benefits over exhaustive approach still hold. A peak energy efficiency of 5.6GOPS/mW of the programmable classification processor validates our idea about having efficiency and flexibility in one chip. The 16-core SDR processor, on the other hand, features domain-specific kernels, flexible ISA control and multi-scale interconnects. It achieves a peak energy efficiency of 13.1GOPS/mW (76fJ/OP) at 415mV, 25MHz, and a peak performance of 1.17TOPS at 1V, 500MHz, showing >2.4× higher energy efficiency than state-of-the-art communication chip multiprocessors, and closing the gap with functionally-equivalent ASICs to within 2.6×.

6.1 Research Contributions

Primary contributions:

- Development of a three-step (coarse-fine-residual) estimation algorithm for energy-efficient and real-time blind signal classification in a 500MHz wideband channel. The proposed hierarchical estimation framework achieves a 59× energy saving

compared to an exhaustive frequency search approach.

- Development of a algorithm-architecture co-design methodology to decide the proper processing strategy and the suitable flexible architecture to support multi-algorithm classification tasks. Combined with aggressive voltage scaling and parallelism, a 3.1× higher energy efficiency is achieved for blind signal feature extraction.

- Development of a dynamic parallelism and frequency scaling framework that keeps the circuit's efficiency always near the optimal value regardless of the bandwidth of the incoming blind signals. Together with the multi-core dynamic scheduling and power gating to improve the hardware utilization, the proposed multi-signal blind classification processor is expected to achieve another 2× efficiency improvement over the single-signal blind classifier.

- Development of a 2×2 kernel structure that matches the domain-specific processing of SDR workload. This efficient datapath structure balances the flexibility and efficiency among various SDR tasks, such as equalization, filtering, matrix decomposition and even multi-antenna sphere decoding.

- Development of a flexible instruction set architecture that dramatically saves the energy and control overhead compared to traditional processor control circuits. The flexible ISA also offers the freedom to define new instructions, enables seamless architectural transformation and task-level hand-over, and can potentially save the program runtime with the the help of programmable state machine.

Benchmarking result shows that the flexible control circuits saves 6.4× and 3.3× of ISA decoder and instruction memory energy, respectively. An overall energy saving of 1.8× is observed from the 4×4 CORDIC-based matrix decomposition.

- Development of a 16-core DSP processor that achieves ≥2.4× higher energy efficiency than state-of-the-art communication processors, and closes the efficiency gap toward functionally-equivalent ASICs to within 2.6×.

- Development of a programming model for the 16-core universal DSP. The model is based on profiling the data- and control-flow mapping of each task and develop a custom assembler to efficiently define the ISA, allocate the BCE resources, and route the interconnects.

Other contributions:

- Development of a multi-core integration strategy to improve the silicon utilization without losing the performance (i.e. loosing the timing constraints). As demonstrated in the 16-core universal DSP, a 95% of silicon utilization is achieved with 500MHz system operating frequency.

- Design of a high-performance 6T memory bit-cell and the configuration system for the multi-granularity FPGA.

- Development of a hybrid-algorithm signal detection, the sphere-MCMC scheme, with energy-efficient VLSI architecture for better decoding performance and lower computing complexity than single-algorithm approaches.

6.2 Future Work

The research on having both flexibility and efficiency in one chip by *design* still requires a lot of evaluation and continued studies. One urgent problem to resolve in the short term is to develop a suitable compiler to map the functions to the coarse-grained reconfigurable hardware. Existing compilers are all designated for RISC processors, so the compilation results cannot be applied. To evaluate the efficiency of the chips in this dissertation, an in-house assembler is developed to generate the machine code. However, it takes a huge amount of human time to manually write the assembly code. Such inefficiency is so far the biggest obstacle to impede the designers from thinking about architectural innovation.

On the other hand, although this work has demonstrated some effective techniques for small-scale designs, more careful analysis is necessary to validate their applicability to SoC level. Theoretical formulation to quantify the design tradeoff can be one direction to justify the value of the proposed techniques. A complete system-level integration can also be a potential direction to demonstrate a truly energy-efficient and flexible radio. Such system should embed more advanced on-chip network and master controllers to deploy the processing threads, coordinate and exploit the results from the processing blocks to initialize the corresponding dependent functions and tasks. Dedicated accelerators are also necessary for a flexible-radio system, which includes the novel non-binary low-density-parity-check (LDPC) decoder, the flexible demodulation FFT kernels supporting multiple antennas, and more. It is also important to understand

the synchronization issues between the physical-layer hardware and the MAC-layer software, so the entire down-/up-link signal chain can be jointly optimized with the goal to maximize energy efficiency and flexibility to adapt to frequent design changes and software updates.

Although the proposed techniques are demonstrated in the domain of wireless communications, they are actually potentially promising to other domains of computing. As a result, a joint algorithm-architecture co-optimization from multiple different domains can be another potential research direction. One example goes to the multimedia and the image signal processing, in that most of the algorithms are still based on conventional matrix operations, just like the case for wireless communications. Considering the era of heterogeneous SoCs in the future, such flexible architectures with compiler support systems for multiple application domains can be promising to meet the next-generation requirements.

REFERENCES

[1] O. Anjum *et al.*, "State-of-the-Art Baseband DSP Platforms for Software-Defined Radio: A Survey," *EURASIP Journal on Wireless Communication and Networking*, Vol. 5, 2011.

[2] Y. Lin *et al.*, "SODA: A High-Performance DSP Architecture for Software-Defined Radio," *IEEE Micro*, vol. 27 no. 1, pp. 114-123, Jan. 2007.

[3] M. Woh *et al.*, "From SODA to Scotch: the Evolution of a Wireless Baseband Processor," in *proc. IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pp. 8-12, Nov. 2008.

[4] D. Liu *et al.*, "Bridging Dream and Reality: Programmable Baseband Processors for Software-Defined Radio," *IEEE Commun Magazine*, vol. 47, pp. 134-140, Sep. 2009.

[5] Tensilica Inc. [Online]. http://www.tensilica.com/uploads/pdf/connx_bbe.pdf.

[6] nVIDIA Inc., "nVIDIA SDR Technology: The Modem Innovation Inside nVIDIA i500 and Tegra 4i," [Online]. http://www.nvidia.com/docs/IO/116757/NVIDIA_i500_whitepaper_FINALv3.pdf.

[7] C. Jalier *et al.*, "A Homogeneous MPSoC with Dynamic Task Mapping for Software Defined Radio," in *proc. IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pp. 345-350, Jul. 2010.

[8] B. Noethen *et al.* "A 105GOPS 36mm^2 Heterogeneous SDR MPSoC with Energy-Aware Dynamic Scheduling and Iterative Detection-Decoding for 4G in 65nm CMOS," in *proc. IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pp. 188-189, Feb. 2014.

[9] B. Mei *et al.*, "ADRES: An Architecture with Tightly-Coupled VLIW Processor and Coarse-Grained Reconfigurable Matrix," in *proc. International Conference on Field-Programmable Logic and Applications*, pp. 61-70, Jan. 2003.

[10] A. Poon, "An Energy-Efficient Reconfigurable Baseband Processor for Wireless Communications," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 15, no. 3, pp. 319-327, Jan. 2007.

[11] G. K. Rauwerda *et al.*, "Towards Software-Defined Radios using Coarse-Grained Reconfigurable Hardware," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 16, no. 1, pp. 3-13, Jan. 2008.

[12] T. Suzuki *et al.*, "High-Throughput, Low-Power Software-Defined Radio Using Reconfigurable Processors," *IEEE Micro*, vol. 31 no. 6, pp. 19-28, Oct. 2011.

[13] "The XG Vision." [Online]. http://www.ir.bbn.com/ ramanath/pdf/rfc_vision.pdf.

[14] D. Čabrić *et al.*, "Implementation issues in spectrum sensing for cognitive radios," in *Proc. 38th Asilomar Conf. Signals, Systems, and Computers (ASILOMAR)*, vol. 1, Nov. 2004, pp. 772-776.

[15] J. Mitola III, "Cognitive Radio: an Integrated Agent Architecture for Software-Defined Radio," *PHD thesis, KTH Royal Institute of Technology, Stockholm, Sweden,*, 2000.

[16] J. Cong, G. Reinman, A. Bui and V. Sarkar, "Customizable Domain-Specific Computing," *IEEE Design and Test of Computers*, vol. 28, no. 2, pp. 6-15, Mar. 2011.

[17] P. S. Zuchowski *et al.*, "A Hybrid ASIC and FPGA Architecture," in *proc. IEEE/ACM Int. Conf. on Computer Aided Design (ICCAD)*, pp. 187-194, Nov. 2002.

[18] J. Lee *et al.*, "Interference Mitigation via Joint Detection," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 6, pp. 1172-1184, Jun. 2011.

[19] D. Bai *et al.*, "Near ML Modulation Classification," *Proc. IEEE VTC*, Fall 2012.

[20] O. Dobre *et al.*, "Survey of Automatic Modulation Classification Techniques: classical approaches and new trends," *IET Commun.*, vol. 1, no. 2, pp. 137-156, Apr. 2007.

[21] S. Moshavi, "Multi-user Detection for DS-CDMA Communications," *IEEE Commun. Mag.*, vol. 34, no. 10, pp. 124-136, Oct. 1996.

[22] B. Ramkumar, "Automatic Modulation Classification for Cognitive Radios using Cyclic Feature Detection," *IEEE Circuits Syst. Mag.*, vol. 9, pp. 27-45, 2009.

[23] V. Madisett, "Wireless, Networking, Radar, Sensor Array Processing, and Nonlinear Signal Processing," *2nd Edition, The Digital Signal Processing Handbook*, Fig. 21.2, CRC Press, Nov. 2009.

[24] C. C. Wang, F.-L. Yuan, T.-H. Yu, and D. Marković, "A Multi-Granularity FPGA with Hierarchical Interconnects for Efficient and Flexible Mobile Computing," in *proc. IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pp. 460-461, Feb. 2014.

[25] F. F. Digham *et al.*, "On the Energy Detection of Unknown Signals Over Fading Channels," *IEEE Trans. Commun.*, vol. 55, no. 1, pp. 21-24, Jan. 2007.

[26] A. Taherpour *et al.*, "Wideband Spectrum Sensing in Unknown white Gaussian Noise," *IET Commun.*, vol. 2, no. 6, pp. 763-771, Jul. 2008.

[27] H. Sun *et al.*, "Wideband Spectrum Sensing for Cognitive Radio Networks: a Survey," *IEEE Wireless Commun.*, vol. 20, no. 2, pp. 74-81, Apr. 2013.

[28] H. Sun *et al.*, "How Much Time is Needed for Wideband Spectrum Sensing?," *IEEE Trans. Wireless Commun.*, vol. 8, no. 11, pp. 5466-5471, Nov. 2009.

[29] K. Hossain *et al.*, "Wideband Spectrum Sensing for Cognitive Radios With Correlated Subband Occupancy," *IEEE Signal Process. Lett.*, vol. 18, no. 1, pp. 35-38, Jan. 2011.

[30] S. Rodriguez-Parera *et al.*, "Spectrum Sensing over SIMO Multi-Path Fading Channels Based on Energy Detection," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Non. 2008, pp. 1-6.

[31] P. Paysarvi-Hoseini *et al.*, "Optimal Wideband Spectrum Sensing Framework for Cognitive Radio Systems," *IEEE Trans. Signal Process.*, vol. 59, no. 3, pp. 1170-1182, Mar. 2011.

[32] D. Čabrić *et al.*, "Physical Layer Design Issues Unique to Cognitive Radio Systems," in *Proc. IEEE 16th Int. Symp. Personal, Indoor and Mobile Radio Commun. (PIMRC)*, vol. 2, Sep. 2005, pp. 759-763.

[33] D. Čabrić *et al.*, "Spectrum Sharing Radio," *IEEE Circuits Syst. Mag.*, vol. 6, no. 2, pp. 30-45, Jul. 2006.

[34] T. Yucek *et al.*, " A Survey of Spectrum Sensing Algorithms for Cognitive Radio Applications," *IEEE Commun. Surveys Tuts.*, vol. 11, no. 1, pp. 116-130, First Quarter 2009.

[35] W. A. Gardner *et al.*, "Characterization of Cyclostationary Random Signal Processes," *IEEE Trans. Inf. Theory*, vol. 21, no. 1, pp. 4-14, Jan. 1975.

[36] W. A. Gardner, "Signal interception: a unifying theoretical framework for feature detection," *IEEE Trans. Commun.*, vol. 36, no. 8, pp. 897-906, Aug. 1988.

[37] Z. Tian *et al.*, "Cyclic Feature Detection With Sub-Nyquist Sampling for Wideband Spectrum Sensing," *IEEE J. Sel. Topics Signal Process.*, vol. 6, no. 1, pp. 58-69, Feb. 2012.

[38] M. Bkassiny *et al.*, "Wideband Spectrum Sensing and Non-Parametric Signal Classification for Autonomous Self-Learning Cognitive Radios," *IEEE Trans. Wireless Commun.*, vol. 11, no. 7, pp. 2596-2605, Jul. 2012.

[39] W.-L. Chin *et al.*, "Features Detection Assisted Spectrum Sensing in Wireless Regional Area Network Cognitive Radio Systems," *IET Commun.*, vol. 6, no. 8, pp. 810-818, May 2012.

[40] N. Kim, *et al.*, "DSP-based Hierarchical neural network modulation signal classification," *IEEE Trans. Neural Netw.*, vol. 14, no. 5, pp. 1065-1071, Sep. 2003.

[41] J. Xu *et al.*, "Software-Defined Radio Equipped with Rapid Modulation Recognition," *IEEE Trans. Veh. Technol.*, vol. 59, no. 4, pp. 1659-1667, May 2010.

[42] J. G. Proakis *et al.*, "Digital Signal Processing," *Pearson Prentice Hall*, 2007.

[43] T.-H. Yu, "Energy-Efficient VLSI Signal Processing for Wideband Spectrum Sensing in Cognitive Radios," *Ph.D. Dissertation, University of California, Los Angeles*, pp. 102-134, Jun. 2013.

[44] T.-H. Yu, O. Sekkat, S. Rodriguez-Parera, D. Marković, and D. Čabrić, "A Wideband Spectrum-Sensing Processor with Adaptive Detection Threshold and Sensing Time," *IEEE Trans. Circuits Syst. I: Reg. Papers*, vol. 58, no. 11, pp. 2765-775, Nov. 2011.

[45] E. Hogenauer, "An Economical Class of Digital Filters for Decimation and Interpolation," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 29, no. 2, pp. 155-162, Apr. 1981.

[46] F.-L. Yuan, T.-H. Yu, and D. Marković, "A 500MHz Blind Classification Processor for Cognitive Radios in 40nm CMOS," in *proc. IEEE International Symposium on VLSI Circuits (VLSI)*, paper 8.3, June 2014.

[47] M. Seok *et al.*, "A 0.27V 30MHz 17.7nJ/Transform 1024-pt Complex FFT Core with Super-Pipelining," *ISSCC Dig. Tech. Papers*, pp. 342-343, Feb. 2011.

[48] T.-H. Yu, C.-H. Yang, D. Cabrić, and D. Marković, "A 7.4-mw 200-ms/s Wideband Spectrum Sensing Digital Baseband Processor for Cognitive Radios," *IEEE Journal of Solid-State Circuits (JSSC)*, vol. 47, no. 9, pp. 2235-2245, Sept. 2012.

[49] Miao Shi, A. Laufer, Y. Bar-Ness, and Wei Su, "Fourth-Order Cumulants in Distinguishing Single Carrier from OFDM signals," in *Proc. IEEE MILCOM*, San Diego, CA, USA, Nov. 17-19, 2008.

[50] E. Rebeiz, P. Urriza, and D. Čabrić, "Experimental Analysis of Cyclostationary Detectors Under Cyclic Frequency Offsets," in *proc. IEEE Asilomar Conf. Signals Syst.*, 2012.

[51] P. Ciblat and E. Serpedin, "A Fine Blind Frequency Offset Estimator for OFDM/OQAM systems," *IEEE Trans. Signal Process.*, vol. 52, no. 1, pp. 291-296, Jan. 2004.

[52] E. Rebeiz and D. Čabrić, "Low Complexity Feature-based Modulation Classifier and its Non-Asymptotic Analysis, in *proc. IEEE GLOBECOM*, Dec. 2011.

[53] G. Burel, C. Bouder, and O. Berder, "Detection of Direct Sequence Spread Spectrum Transmissions without Prior Knowledge," in *Proc. IEEE GLOBECOM*, Nov. 2001.

[54] W. A. Gardner, "Statistical Spectral Analysis: A Non-Probabilistic Theory," *Prentice-Hall*, 1988.

[55] C. C.Wang, C. Shi, R. W. Brodersen, and D. Marković "An Automated Fixed-point Optimization Tool in Matlab XSG/SynDSP Environment," *ISRN Signal Process.*, 2011, Art. ID 414293.

[56] K.-H. Lin, "Design of a Baseband Receiver for DVB-T Standard," Master Thesis, National Taiwan University, Taipei, 2004.

[57] A. Wenzler and E. Luder, "New Structures for Complex Multipliers and Their Noise Analysis," in *proc. IEEE Int. Symp. Circuits Syst.*, 1995.

[58] J. Rabaey, A. Chandrakasan, and B. Nikolić, "Digital Integrated Circuits: A Design Perspective," *Prentice-Hall*, 2003.

[59] T. Kurafuji *et al.*, "A Scalable Massively Parallel Processor for Real-Time Image Processing," *IEEE J. Solid-State Circuits*, vol. 46, no. 10, pp. 2363-2373, Oct. 2011.

[60] K. Mohammed, "A MIMO Decoder Accelerator for Next Generation Wireless Communications," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 18, no. 11, pp. 1544-1555, Nov. 2010.

[61] A. Poon, "An Energy-Efficient Reconfigurable Baseband Processor for Wireless Communications," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 15, no. 3, pp. 319-327, Jan. 2007.

[62] C. Zhang *et al.*, "Design of Coarse-Grained Dynamically Reconfigurable Architecture for DSP Applications," in *proc. International Conference on Reconfigurable Computing and FPGAs*, pp. 338-343, Dec. 2009.

[63] Z. Yu *et al.*, "An 800MHz 320mW 16-Core Processor with Message-Passing and Shared-Memory Inter-Core Communication Mechanisms," in *proc. ISSCC Dig. Tech. Papers*, pp. 64-65, Feb. 2012.

[64] P. Ou *et al.*, "A 65nm 39GOPS/W 24-Core Processor with 11Tb/s/W Packet-Controlled Circuit-Switched Double-Layer Network-on-Chip and Heterogeneous Execution Array," in *proc. ISSCC Dig. Tech. Papers*, pp. 56-57, Feb. 2013.

[65] D. Truong *et al.*, "A 167-Processor Computational Platform in 65nm CMOS," *IEEE J. Solid-State Circuits*, vol. 44, no. 4, pp. 1130-1144, Apr. 2009.

[66] H. Zhang *et al.*, "A 1-V Heterogeneous Reconfigurable DSP IC for Wireless Baseband Digital Signal Processing," *IEEE J. Solid-State Circuits*, vol. 35, no. 11, pp. 1697-1704, Nov. 2000.

[67] Z. Yu *et al.*, "AsAP: An Asynchronous Array of Simple Processors," *IEEE J. Solid-State Circuits*, vol. 43, no. 3, pp. 695-705, Mar. 2008.

[68] F. Sheikh *et al.*, "A 1-190MSample/s 8-64 Tap Energy-Efficient Reconfigurable FIR Filter for Multi-Mode Wireless Communication," in *proc. Symp. VLSI Circuits Dig.*, pp. 207-208, Jun. 2010.

[69] M. Shabany *et al.*, "A 0.13μm CMOS 655Mb/s 4$\times$4 64-QAM K-Best MIMO Detector," in *proc. ISSCC Dig. Tech. Papers*, pp. 256-257, Feb. 2009.

[70] M. Shabany *et al.*, "A Low-Latency Low-Power QR-Decomposition ASIC Implementation in 0.13μm CMOS," *IEEE Trans. Circuits Syst. I: Reg. Papers*, vol. 60, no. 2, pp. 327-340, Feb. 2013.

[71] F.-L. Yuan, T.-H. Yu, and D. Marković, "A 500MHz Blind Classification Processor for Cognitive Radios in 40nm CMOS," in *proc. IEEE International Symposium on VLSI Circuits (VLSI)*, paper 8.3, June 2014.

[72] D. Markovic *et al.*, "Power and Area Minimization for Multidimensional Signal Processing," *IEEE J. Solid-State Circuits*, vol. 42, no. 4, pp. 922-934, Apr. 2007.

[73] C. Chang *et al.*, "BEE2: A High-End Reconfigurable Computing System," *IEEE Des. Test. Comput.*, vol. 22, no. 2, pp. 114-125, Mar. 2005.

[74] C.-H. Yang *et al.*, "A Flexible DSP Architecture for MIMO Sphere Decoding," *IEEE Trans. Circuits Syst. (I)*, vol. 56, no. 10, pp. 2301-2314, Oct. 2009.

[75] W. R. Davis *et al.*, "A Design Environment for High Throughput, Low Power Dedicated Signal Processing Systems," *IEEE J. Solid-State Circuits*, vol. 37, no. 3, pp. 420-431, Mar. 2002.

[76] S. M. Mishra *et al.*, "A Real Time Cognitive Radio Testbed for Physical and Link Layer Experiments," in *IEEE Proc. 1st Int. Symp. Dynamic Spectrum Access Networks (DySPAN)*, Nov 2005, pp. 562-567.

[77] VITA, [Online]. http://www.vita.com/fmc.html

[78] Xilinx, "I/O Design Flexibility with the FPGA Mezzanine Card (FMC)," white paper 315 (WP315), ver. 1, Aug. 2009, [Online]. http://www.xilinx.com/support/documentation/white_papers/wp315.pdf

[79] Samtec, "FMC VITA57 connector datasheet," [Online]. http://www.samtec.com/documents/webfiles/Cpdf/ASP-134488-01-mkt.pdf

[80] Xilinx, “Xilinx Kintex-7 FPGA KC705 Evaluation Kit,” [Online]. http://www.xilinx.com/products/boards-and-kits/EK-K7-KC705-G.htm

[81] Xilinx, “ChipScope Pro Software and Cores,” user guide 029, ver. 14.3, Oct. 2012, [Online]. http://www.xilinx.com/support/documentation/sw_manuals/xilinx14_4/chipscope_pro_sw_cores_ug029.pdf

[82] Xilinx, “7 Series FPGAs SelectIO Resources,” user guide 471, ver. 1.4, May 2014, [Online]. http://www.xilinx.com/support/documentation/user_guides/ug471_7Series_SelectIO.pdf

[83] Xilinx, “KC705 Power Bus Reprogramming,” tutorial, 2013, [Online]. http://www.xilinx.com/Attachment/KC705_Power_Controllers_Reprogramming_Steps.pdf