Designing Flexible, Energy Efficient and Secure Wireless Solutions for the Internet of Things

by

Yajing Chen

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Computer Science and Engineering)
in the University of Michigan
2017

Doctoral Committee:

Professor Trevor N. Mudge, Co-Chair
Assistant Professor Hun Seok Kim, Co-Chair
Professor David T. Blaauw
Assistant Professor Ronald G. Dreslinski Jr.
Associate Professor Thomas F. Wenisch

Yajing Chen
yajchen@umich.edu
ORCID iD: 0000-0002-8767-5945

To my parents and Shengshuo.

ACKNOWLEDGMENTS

I would like to thank my advisors–Professor Trevor Mudge and Professor Hun Seok Kim.

Their support and guidance has been invaluable. They were there to suggest research directions and to listen to my proposed solutions.

I would like to acknowledge Professor Ronald Dreslinski and Professor David Blaauw for also advising me during my studies and giving me an appreciation of the implementation trade-offs that are necessary consequences of my ideas. I would also like to acknowledge Professor Thomas Wenisch, who in his role as a member of my committee, provided an important wider perspective on my work, and Professor Achilleas Anastasopoulos for introducing me to research in communications.

To all my lab-mates: Cao Gao, Nilmini Aberaytne, Johann Hauswald, Byoungchan Oh, Jonathan Beaumont, Qi Zheng, Anthony Gutierrez, Yiping Kang, Joseph Pusdesris, Thomas Manville, Michael Cieslak, Dong-Hyeon Park, thank you for many interesting conversations and making my stay in lab an enjoyable experience.

To my colleagues in the department: Shengshuo Lu, Yiqun Zhang, Yang Liu, Supreet Jeloka, Amlan Nayak, Paul Myers, Cheng Fu, thank you for helping me solving research problems, discussing ideas, editing paper and being my friends.

I would also like to thank my parents and family for understanding, supporting and encouraging me.

Finally, I would like to thank Trev for taking me into the PhD program, where I found my lifetime partner— Mr. Lu.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Designing Flexible, Energy Efficient and Secure Wireless Solutions for the Internet of Things

by

Yajing Chen

Chair: Trevor N. Mudge, Hun Seok Kim

The Internet of Things (IoT) is an emerging concept where ubiquitous physical objects (things) consisting of sensor, transceiver, processing hardware and software are interconnected via the Internet. The information collected by individual IoT nodes is shared among other often heterogeneous devices and over the Internet.

This dissertation presents flexible, energy efficient and secure wireless solutions in the IoT application domain. System design and architecture designs are discussed envisioning a near-future world where wireless communication among heterogeneous IoT devices are seamlessly enabled.

Firstly, an energy-autonomous wireless communication system for ultra-small, ultra-low power IoT platforms is presented. To achieve orders of magnitude energy efficiency improvement, a comprehensive system-level framework that jointly optimizes various system parameters is developed. A new synchronization protocol and modulation schemes are specified for energy-scarce ultra-small IoT nodes. The dynamic link adaptation is proposed to guarantee the ultra-small node to always operate in the most energy efficiency mode, given an operating scenario. The outcome is a truly energy-optimized wireless communication system to enable various new applications such as implanted smart-dust devices.

Secondly, a configurable Software Defined Radio (SDR) baseband processor is designed and shown to be an efficient platform on which to execute several IoT wireless standards. It is a custom SIMD execution model coupled with a scalar unit and several architectural optimizations: streaming registers, variable bitwidth, dedicated ALUs, and an optimized reduction network. Voltage scaling and clock gating are employed to further reduce the power, with a more than a 100% time margin reserved for reliable operation in the near-threshold region. Two upper bound systems are evaluated. A comprehensive power/area estimation indicates that the overhead of realizing SDR flexibility is insignificant. The benefit of baseband SDR is quantified and evaluated.

To further augment the benefits of a flexible baseband solution and to address the security issue of IoT connectivity, a light-weight Galois Field (GF) processor is proposed. This processor enables both energy-efficient block coding and symmetric/asymmetric cryptography kernel processing for a wide range of GF sizes (2^m, $m = 2, 3, ..., 233$) and arbitrary irreducible polynomials. Program directed connections among primitive GF arithmetic units enable dynamically configured parallelism to efficiently perform either four-way SIMD GF operations, including multiplicative inverse, or a long bit-width GF product in a single cycle. This demonstrates the feasibility of a unified architecture to enable error correction coding flexibility and secure wireless communication in the low power IoT domain.

CHAPTER 1

Introduction

1.1 Internet of Things (IoT)

The Internet of Things (IoT) envisions a world where ubiquitous physical objects are IP addressable, enabling various connectivity among things or everything, as well as active interactions between things and human beings.

Emerging applications in various areas such as industry, smart home, transportation, environment, health care are powered by IoT technologies as shown in Figure 1.1. For example, cost control, production efficiency and process management are example applications in Industry IoT (IIoT). Smart home applications are typified by home automation system, monitoring and security surveillance, and interactive voice system, etc. IoT solutions for smart city involve a wide range of user cases, including traffic management, urban security, waster management/distribution and so on. Many other areas such as logistics, smart agriculture, health system, are developing rapidly as IoT technologies evolve.

The IoT market has experienced exponential growth throughout the past years. The volume of things will be connecting to the Internet is going to exceed 50 billion [1] by 2020. The new IoT opportunities is estimated to power the industry with approximately $400 billion dollars by 2020 as shown in Fig. 1.2 [1].

1.2 IoT Wireless Communications

Central to the IoT vision is wireless connectivity, which ultimately connects the isolated physical objects together as well as enables data sharing, information exchanging and interactions among things and humans, within the local network and over the Internet.

Similarly to other communication system, IoT wireless communications also consists of multiple layers as shown in Fig. 1.3.

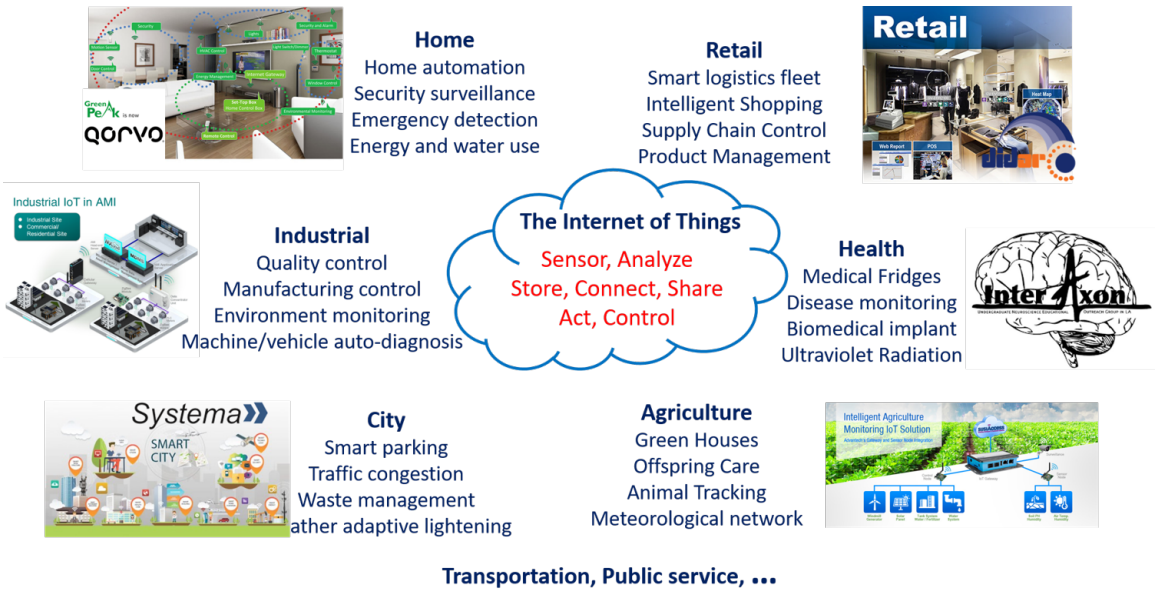


Figure 1.1: IoT Application Areas

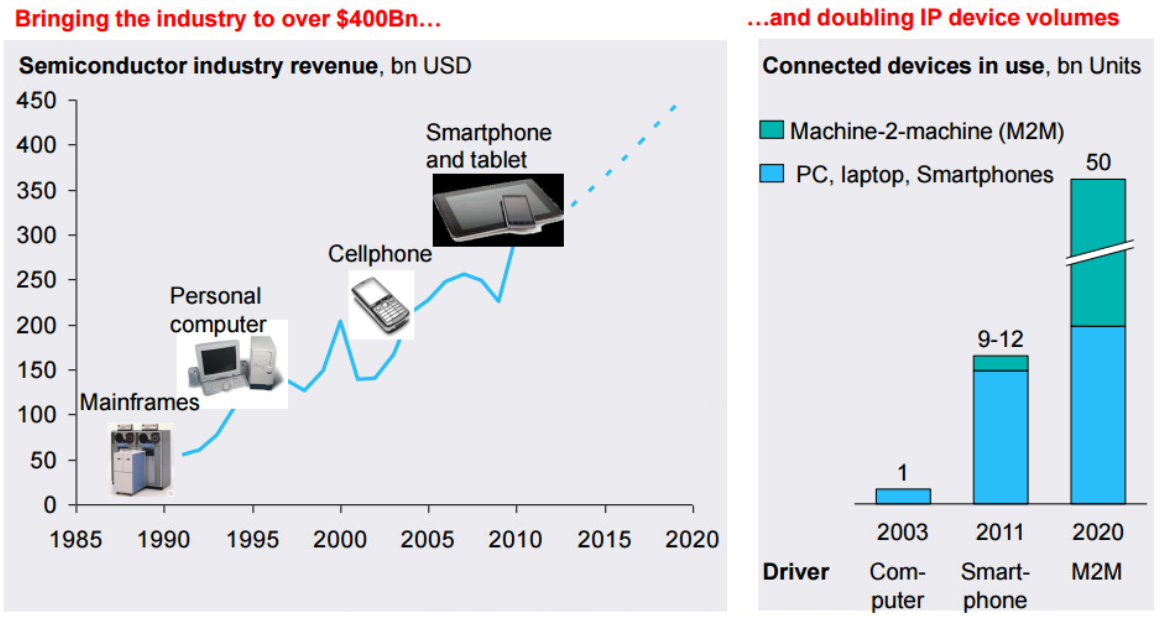


Figure 1.2: IoT brings the industry to over $400 billion [1]

Layer	Services	
Transport Layer (L4)	Connection-oriented model, Reliability: checksum, Flow control, Congestion avoidance, Multiplexing ports	
Network Layer (L3)	IP Layer	Connectionless model IP addressing, Routing
	Adaptation Layer	Address auto-configuration Fragmentation & assembly Header Compression Security: key management
Link Layer (L2)	Packet Switching Hardware Addressing, Physical medium access control Security: encryption, authentication Error detection and control: FEC	
PHY Layer (L1)	Baseband processing	

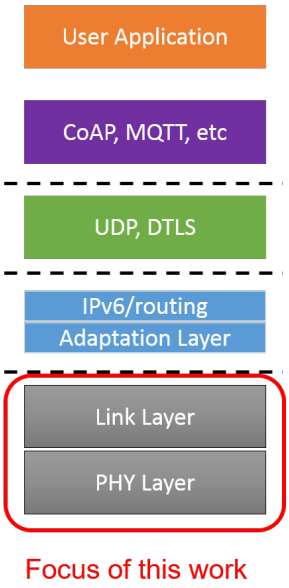


Figure 1.3: IoT communications stack.

The physical layer, also referred as the PHY layer or L1, is the lowest layer in the communications stack. The physical layer is responsible for baseband signal processing, including Modulation, synchronization, FIR, etc. On top of the physical layer, the link layer, also referred as L2, provides several services including packet switching, hardware addressing, physical medium access control, security and forward error correction coding (FEC). The main functionality provided by each layer (below the application layer), including the Network layer, and the Transport layer are listed in the left column in Fig. 1.3. An adaptation layer is necessary in between L2 and the traditional L3. IPv6 over Low power WPAN (6lowpan) is a popular adaptation layer protocol. The adaptation layer is specially designed to address the difference between low power IoT networks and traditional networks, providing services such as packet fragmentation and reassembly, and header compression. This dissertation focuses on system and architecture design for the physical layer, and low level signal processing in L2. To be specific, the low level L2 processing includes forward error correction coding and security schemes (including both data encryption/decryption and authentication processing), which are tightly integrated with baseband processing in the physical layer.

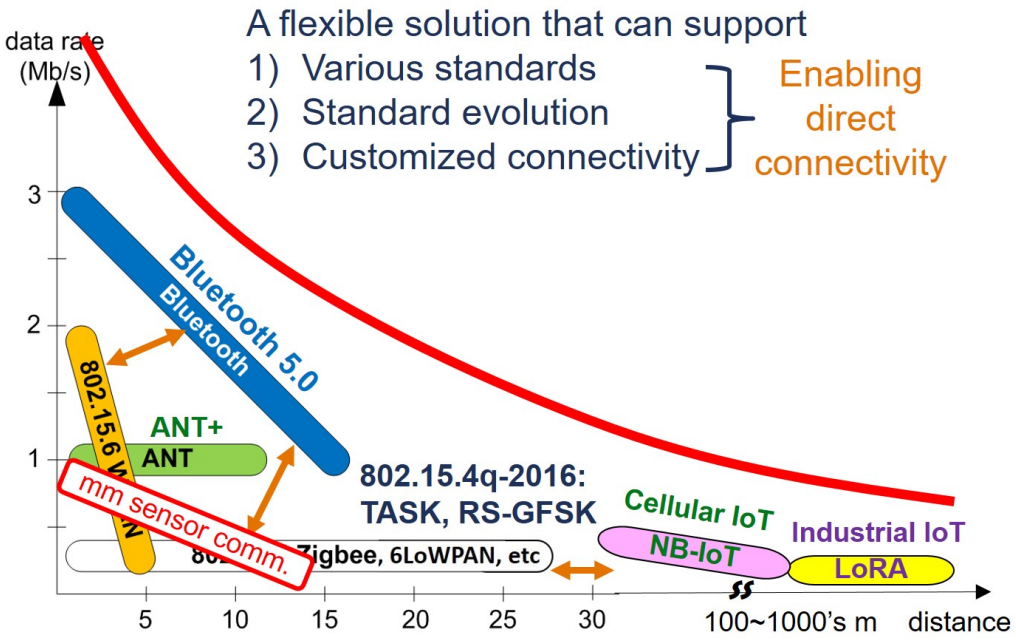


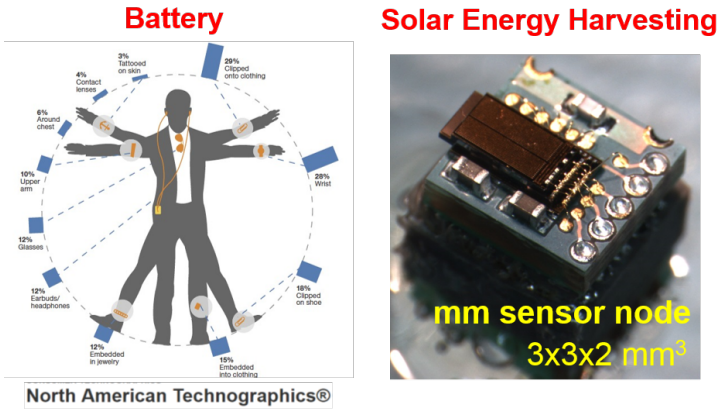
Figure 1.4: Design Principle: Flexibility.

1.3 Design Principles

There are many standards that have already been proposed for IoT such as IEEE 802.15.4 [2], IEEE 802.15.6 [3], and Bluetooth Low Energy (BLE) [4]. The rapid growth in IoT applications brings new demands on wireless communications. The standards are continuously evolving to adopt new technologies. For example, Bluetooth 5.0 [5] released in December last year has increased the maximum data rate of BLE from 1Mbps to 2Mbps. Two alternative PHYs have been added to the 802.15.4 standard in 2016—Ternary ASK (TASK) and Rate-Switching GFSK (RS-GFSK). They are specified in 802.15.4q-2016 [6].

Besides updates to standards, new ones have been developed. For instances, cellular IoT, typified by Narrow Band (NB)-IoT, having a single tone providing 20kbps and a multiple tone with a maximum data rate of 250kbps, have been proposed. In addition, Industrial IoT (IIoT) standards, such as LoRA [7], have come into the picture. They operate at a very low data rate (30-50kbps) but extend range to kilometers.

Today, although there are many different types of connectivity proposed for IoT, these networks are often fragmented. Nodes with different physical layers are unable to directly communicate with each other, because typically devices only have one single physical implementation. In the near future, where a trillion of IoT devices communicating with other heterogeneous devices should not be limited by one specific physical layer solution.



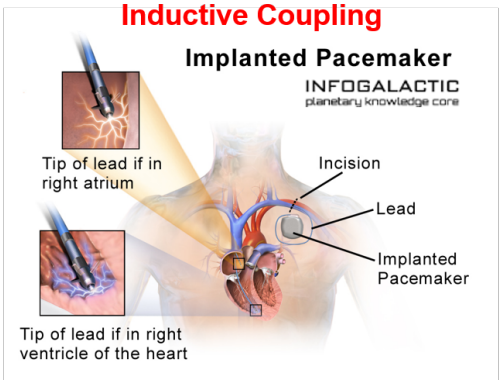


Figure 1.5: IoT devices typically rely on limited energy sources.

Flexibility. Ideally, the IoT devices should have the flexibility, as shown in Fig. 1.4, to support many standards, non-standard, and custom specified communications such as millimeter scale IoT communications which will be elaborated later.

Energy Efficiency. Energy efficiency is always one of the most essential design considerations because the power/energy sources of IoT devices are typically restricted. Example energy sources are illustrated in Fig. 1.5. Most wearable devices are powered by battery, but charging and replacing battery is costly or even infeasible. Energy harvesting is another important energy source for IoT devices, for example, millimeter scale sensor node relies on solar power harvesting. Inductive power suppliers involves wirelessly transferring energy from one device (transmitter or charging station) to another (receiver or portable device). RFID is a typical and a widely adopted use case of inductive coupling. The implanted pacemaker in Fig. 1.5 is another inductive power example. Given the energy limitation, designing an energy efficient solutions is often critical for IoT devices.

It is important to note that flexibility does not necessarily worsen energy efficiency. A flexible wireless solution will add some complexity compared to a single ASIC solution,

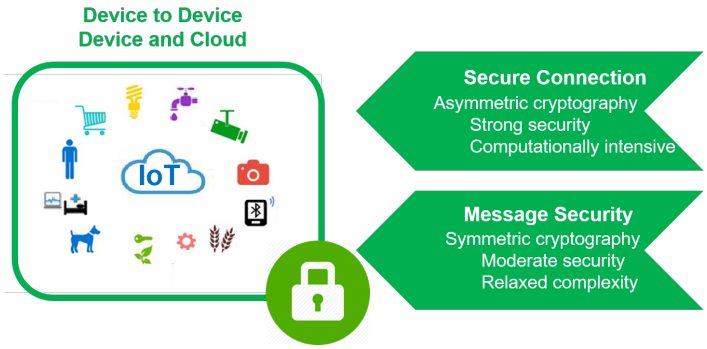


Figure 1.6: Design Principle: Security.

but the operating scenario is dynamic. The flexibility opens up the opportunity to achieve better energy efficiency by selecting an energy optimized configuration point that may be beyond the scope of existing standard modes. This dissertation explores the flexibility in the physical layer, i.e. baseband processing, and error correction coding. The flexibility in baseband processing and information/error correction coding schemes allows the IoT devices to dynamically change multiple system parameters, such as signal bandwidth, modulation parameters, and coding schemes, for different objectives such as higher data rate, better energy efficiency and/or longer distance.

The area/system volume of the IoT devices is another important concern in order to connect to the real world seamlessly and unnoticeably. The wireless flexibility should also be achieved with small area and system volume. This dissertation analyzes the fundamental algorithm in the physical layer and low level signal processing in L2. We show that flexibility can be addressed within a small area by designing architectures that efficiently support basic common computations.

In some applications, such as biomedical implants, where extremely small dimensions are required, none of the standard communications can be adopted and unique challenges due to ultra-small system form factors are introduced. Chapter 2 details a system designed for such ultra-small, ultra-low power wireless communications for IoT devices. Specialized modulation schemes, synchronization protocols, and link adaptation schemes are proposed to optimally utilize the scarce power/energy resources.

Security. For IoT communications, security is an important design consideration. IoT communications, which exchange plaintext through a shared medium, is fundamental insecure. Fig. 1.6 illustrates the security issues addressed in this dissertation. Firstly, a secure connection needs to be established, typically via asymmetric cryptography process, which has strong security but is very computationally intensive. Elliptic Curve Cryptography, as a representative asymmetric cryptography, is widely utilized in the authentication process.

Secondly, the message transmitted over the air needs to be secure, and symmetric cryptography algorithms are usually implemented to encrypt/decrypt messages. They provide a moderate security level with relaxed complexity. AES is one of the most popular symmetric cryptography schemes, which is specified in many IoT standards.

In summary, **flexibility, energy efficiency and security** are essential design principles in the areas of IoT wireless communications. In this dissertation, flexible, energy efficient and secure IoT wireless solutions are presented via a comprehensive system and detailed architecture design.

1.4 Dissertation Outline

This dissertation is organized as follows. In Chapter 2, an energy-autonomous, self-contained wireless communications system designed for ultra-small millimeter-scale, ultra-low power IoT sensor node is presented. In such cases, none of the standard communications can be applied and even convectional communications technique cannot be directly applied.

Chapter 3 and Chapter 4 focus on architecture design for standard compliant IoT platforms. Chapter 3 describes a configurable SDR baseband processor. Several reasons, such as multi-standard scenarios, on-the-fly updates, and graceful data rate/distance trade-offs, can be handled easily by SDR as long as the overhead of SDR can be kept small compared to overall system. We show this is the case for our processor in this chapter. Better energy efficiency can be achieved by operating the SDR at an energy optimal configurations which can be beyond the standard for existing physical layer protocols. Chapter 4 further augments this benefit from baseband flexibility by introducing information coding flexibility in IoT communications. As mentioned in the previous section, in addition to energy efficiency and flexibility, security is also one of the most critical challenges posed for IoT wireless connectivity. The Galois Field (GF) processor proposed in Chapter 4 addresses the security challenges with a unified architecture to perform not only error correction coding but also popular cryptography functions in a finite GF for both data encryption/decryption and authentication processes.

Finally, Chapter 5 provides concluding remarks.

CHAPTER 2

Energy-Autonomous Wireless Communication for Millimeter-Scale Internet-of-Things Sensor Nodes

This chapter presents an energy-autonomous wireless communication system for ultra-small Internet-of-Things (IoT) platforms. In the proposed system, all necessary components including the battery, energy-harvesting solar cells, and the RF antenna are fully integrated within a millimeter-scale form-factor. Designing an energy-optimized wireless communication system for such a miniaturized platform is challenging because of unique system constraints imposed by the ultra-small system dimension. The proposed system targets orders of magnitude improvement in wireless communication energy efficiency through a comprehensive system-level analysis that jointly optimizes various system parameters such as node dimension, modulation scheme, synchronization protocol, RF/analog/digital circuit specifications, carrier frequency, and miniaturized 3D antenna efficiency. We propose a new protocol and modulation schemes that are specifically designed for energy-scarce ultra-small IoT nodes. These new schemes exploit abundant signal processing resources on gateway devices to simplify design for energy-scarce ultra-small sensor nodes. The proposed dynamic link adaptation guarantees that the ultra-small IoT node always operates in the most energy efficient mode for a given operating scenario. The outcome is a truly energy-optimized wireless communication system to enable various classes of new applications such as implanted smart-dust devices.

2.1 Introduction

Ultra-small Internet-of-Things (IoT) sensor nodes with perpetual energy harvesting have come into reality empowered by very-large-scale system integrated (VLSI) circuit innovations [8] – [9] and fabrication technology improvement. Leading into the realistic world of

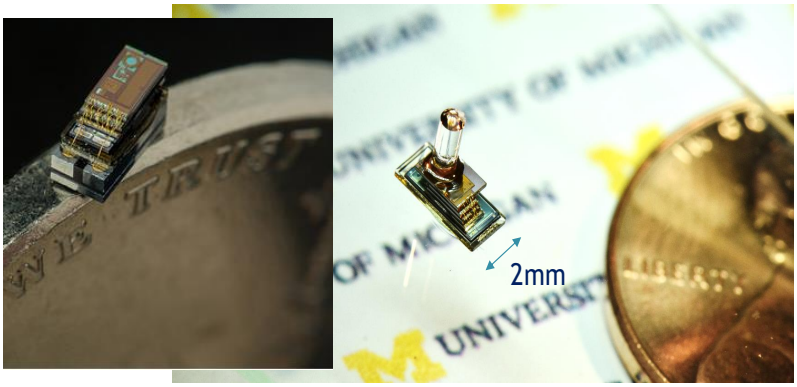


Figure 2.1: Millimeter-scale Michigan Micro Mote (M^3), pressure sensing system (left) and imaging system (right)

'smart dust' [8] – [10], ultra-small (specifically in millimeter-scale) IoT platforms present a wide range of new applications such as biomedical implant [11] [12], security/safety surveillance, infrastructure monitoring [13] [14], and smart building [15], which are all extremely platform size sensitive.

We envision a millimeter-scale general purpose computing platform (Fig. 2.1) [8] – [9] that is small enough to be seamlessly integrated into the real world without being noticed. Recent research in VLSI circuits demonstrate that realizing such an ultra-small system is indeed feasible by integrating a rechargeable battery, a solar energy harvester layer, various sensor (temperature, pressure, imager) layers, and a general purpose processor layer; all within a millimeter-scale form-factor (see Fig. 2.1). The extremely small form-factor of these systems imposes a critical challenge in system power and energy management. As battery replacement is practically infeasible, these systems target energy-autonomous operation, employing a millimeter-scale rechargeable thin-film battery that is continually trickle charged by harvested ambient (e.g., solar) energy [8] – [9].

In millimeter-scale IoT platforms, wireless communication is the dominant factor in overall power/energy consumption [8] – [9] [16] – [17].

The power breakdown of the millimeter-scale, energy-autonomous Michigan Micro Mote (M3) sensor node (Fig. 2.1) [8] – [9] reveals that wireless communication consumes more than 65% of the overall power budget even with aggressive duty cycling. Therefore, enabling energy optimized wireless communication is the most critical issue in prolonging the lifespan of energy-constrained ultra-small IoT platforms and realizing perpetual device operation only powered by energy harvesting.

In this chapter, we present a truly energy-autonomous, fully self-contained wireless communication system that optimally utilizes the scarce energy/power resources available

in ultra-small scale IoT platforms. To achieve this goal, a cross-layer system-level optimization framework is proposed to jointly optimize various system parameters such as modulation scheme, synchronization protocol, RF/analog/digital circuit specifications, data rate, carrier frequency, miniaturized 3D antenna efficiency, etc. The proposed optimizations will be performed under stringent system constraints that are unique to millimeter-scale energy-harvesting IoT platforms. Given a millimeter-scale IoT node dimension, we seek to explore the rich tradeoff space between the miniaturized 3D antenna size, battery capacity, and the energy reservoir capacitor size to achieve the maximum energy efficiency for wireless communication.

We assume that each ultra-small IoT node is paired with a gateway such as a smartphone or an WiFi access point. Direct communication among ultra-small IoT sensor nodes is not the focus of this work. Since the energy/power efficiency of the gateway device is not a primary concern, it is reasonable to assume that the gateway is equipped with abundant resources for advanced signal processing. Exploiting this asymmetry, this work will investigate an energy-optimized communication system for a power-constrained IoT sensor node with very dissimilar transmitter and receiver configurations. The proposed system will demonstrate that powerful signal processing at the gateway can mitigate circuit impairments on the ultra-low power (ULP, $< 100\mu W$) sensor node. Hence, the circuit specification on the sensor node can be significantly relaxed for better energy efficiency, benefiting from the gateway signal processing. A new synchronization protocol is investigated to improve the sensor node receiver energy efficiency by utilizing accurate timing and frequency offset estimation attainable on the gateway. This synchronization scheme eliminates the need for a power-demanding phase-lock-loop (PLL) circuit and an external frequency reference crystal for the millimeter-scale sensor node wireless transceiver.

Conventional modulation schemes, such as on-off-keying (OOK) and binary frequency shift keying (FSK), widely used in prior ULP transceiver works [18] [9] are far from optimal for the proposed system since these conventional simplistic modulation schemes lack the ability to dynamically adapt to various operating scenarios. We propose a new modulation-coding scheme that is specifically designed for millimeter-scale IoT sensor nodes to expand their connectivity in an energy-efficient way. In addition, we will show that gateway guided link adaptation provides an order(s)-of-magnitude improvement in data rate and/or link distance by automatically adjusting the modulation-coding scheme and other modulation parameters on-the-fly for the millimeter-scale IoT sensor node.

This chapter is organized as follows. Section 2.2 discusses the energy-autonomous millimeter-scale communication system design considerations. A new synchronization protocol is discussed in Section 2.3. Section 2.4 provides mathematical models of the

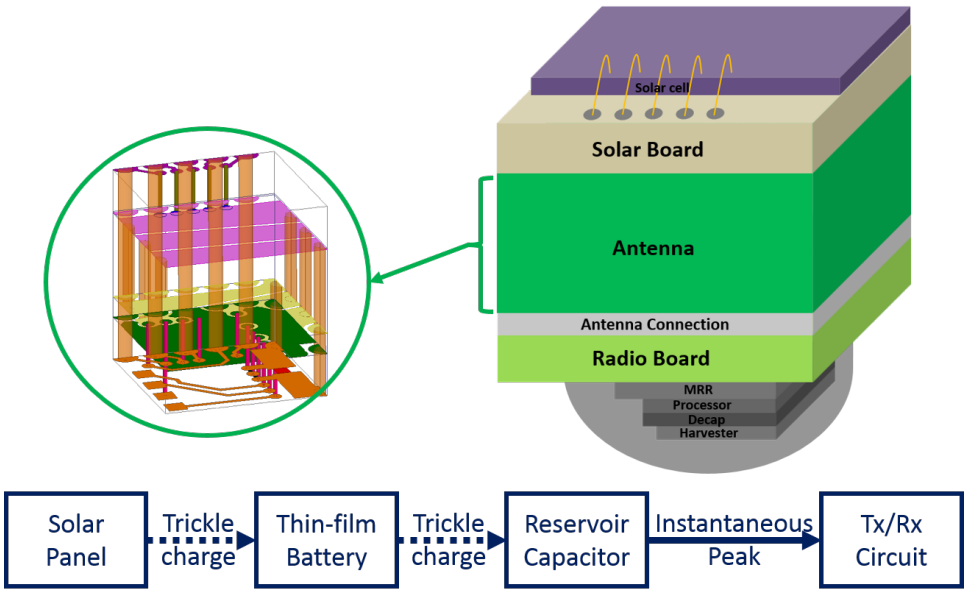


Figure 2.2: Millimeter-scale stacked system diagram (up-right) and 3D magnetic antenna prototype (up-left). The flow of energy-autonomous operations.

modulation scheme, coding, synchronization, energy consumption, and data rate of the proposed system. In Section 2.5, we mathematically define the dynamic link adaptation problem for various objective functions, and quantify the optimum results and the impact of different system configurations. Section 2.6 concludes this chapter.

2.2 System Design

In this section, we will introduce system constraints of the energy-autonomous millimeter-scale communication system. The proposed system design is driven by the objective of achieving the maximum energy efficiency, longest link distance, and/or highest data rate. Critical system design decisions such as the antenna type and modulation scheme will be justified in this section.

2.2.1 Millimeter-Scale Wireless Communication System Constraints

The proposed energy-autonomous wireless system configuration is depicted in Fig. 2.2. A cubic shape form-factor is considered to minimize the system volume and to constrain the length of the largest dimension. The transceiver circuits, thin-film battery, and other discrete components including the energy reservoir capacitor (C_{res}) are sandwiched between the solar energy harvester panel and the millimeter-scale 3D antenna. A stacked layer approach introduced in [8] – [9] shall be applied to integrate multiple functional layers such

as a processor layer and sensor layers on top of each other with minimal area overhead.

The millimeter-scale dimension (largest dimension is less than $5mm$) imposes several unique constraints for energy-autonomous wireless systems. For example, the peak current of a millimeter-scale thin film battery cannot exceed $\approx 100\mu A$ [8] – [9] because of the internal battery resistance. We utilize a discrete $\leq 1\mu F$ capacitor (see Fig. 2.2) as the energy reservoir (or energy buffer) to address the mismatch between the instantaneous peak power requirement for transceiver operation and the peak battery current limitation. The impact of this energy reservoir capacitor will be analyzed in Section IV.C. The solar energy harvesting layer is exposed on one side of the stack as shown in Fig. 2.2. It is reasonable to expect 10s of μW power harvesting per mm^2 [8] – [9] [19] from the state-of-the-art millimeter-scale solar cells and energy harvesting IC. The rechargeable thin-film battery and the energy reservoir capacitor are constantly trickle charged by the harvested solar energy. Energy-autonomous operation is satisfied when the average power consumption for wireless communication is contained below the harvested power level, while the energy reservoir capacitor buffers the instantaneous peak power demand.

2.2.2 Millimeter-Scale 3D Antenna Modeling and Design

As the proposed system has unconventionally small dimensions, analyzing the antenna efficiency is a necessary step to design an optimized communication system. In particular, we compare two possible options—electric and magnetic dipoles—for millimeter-scale 3D antenna design, and model their efficiency as a function of system dimension and carrier frequency. The goal of this analysis is to determine the optimal antenna topology and operating carrier frequency for millimeter-scale wireless systems.

Electrically-small antennas radiate as electric and/or magnetic dipoles. The equivalent circuit of a small dipole antenna comprises radiation resistance (R_{rad}), loss resistance (R_{loss}), and capacitance (for electric dipole) / inductance (for magnetic dipole), as tabulated in Table 2.1 [20] [21]. Small 'electric' dipoles (straight wires) have capacitive input impedances and relatively high radiation efficiencies (η_e). On the contrary, small 'magnetic' dipoles (circular loops) have inductive input impedances, and theory indicates much lower radiation efficiencies (η_m); e.g., $\eta_e = 0.62 \gg \eta_m = 0.01$ for millimeter scale antennas with $r_{ant} = 2.5mm$, $l_{ant} = 5mm$, and carrier frequency $f_c = 1GHz$.

In the proposed system, a series lumped element will be used to make each of the two dipoles resonant. We will not concern ourselves with matching the antenna to a specific impedance because, for a narrowband system, inter-symbol interference from impedance mismatched RF reflection is a secondary concern. Capacitive (inductive) nature of an elec-

Table 2.1: Antenna Characteristics and Efficiency Modeling

	Electric Dipole	Magnetic Dipole
Constants, Assumptions	k: wave number, c: the speed of light $d_w = 0.2mm$:wire diameter μ_0: permeability of free space $\rho = 17.2n\Omega/m$: resistivity of copper wire	
Antenna Dimension	l_{ant}: length of the dipole	r_{ant}: radius of the loop
Radiation Resistance, R_{rad}	$20\left(\frac{kl_{ant}}{2}\right)^2$	$20\pi^2(kr_{ant})^4$
Loss Resistance, R_{loss}	$\frac{l_{ant}}{2\pi d_w}\sqrt{\frac{kc\mu_0\rho}{2}}$	$\frac{2r_{ant}}{d_w}\sqrt{\frac{kc\mu_0\rho}{2}}$
Capacitance / Inductance	$C^e = \frac{l_{ant}}{240\Omega\ c\left(\ln\left(\frac{l_{ant}}{d_w}\right)-1\right)}$	$L^m = \mu_0 r_{ant}\left(\ln\left(\frac{16r_{ant}}{d_w}\right)-2\right)$
Radiation Efficiency	$\eta_e = \frac{R_{rad}}{R_{rad}+R_{loss}}$	$\eta_m = \frac{R_{rad}}{R_{rad}+R_{loss}}$
Rad. eff. Inc. lumped comp. loss	$\hat{\eta}_e = \frac{R_{rad}}{R_{rad}+R_{loss}+R_{L^e}}$	$\hat{\eta}_m = \frac{R_{rad}}{R_{rad}+R_{loss}+R_{C^m}}$

tric (magnetic) dipole antenna requires lumped inductor (capacitor) to make the antenna resonate. Since lumped capacitors exhibit much higher quality factor than inductors, the antenna efficiency, η_e and η_m, of the electric and magnetic dipole antenna should be reevaluated considering realistic quality factors of lumped series capacitors and inductors.

The required inductance for an electric dipole is $L^e = 1/(\omega^2 C^e)$ whereas a magnetic dipole resonates with a series capacitance $C^m = 1/(\omega^2 L^m)$, where ω is the angular frequency. It is reasonable to assume that the quality factor of the inductor and capacitor is $Q_{L^e} = (\omega L^e)/R_{L^e} = 50$ and $Q_{C^m} = 1/(\omega C^m R_{C^m}) = 250$, respectively. These values are typical among surface mount inductors and capacitors in the carrier frequency range of interest, $f_c < 6GHz$.

The radiation efficiency for electric and magnetic dipoles is a function of antenna dimension and carrier frequency as shown in Table 2.1 and Fig. 2.3. Considering the realistic quality factor of lumped elements, magnetic dipoles can lead to higher overall radiation efficiencies in millimeter-scale designs outperforming electric dipoles ($\hat{\eta}_m > \hat{\eta}_e$) as shown in Fig. 2.3 bottom, given $f_c \leq 5GHz$. Notice the opposite is true ($\eta_m \ll \eta_e$) if quality factors of lumped elements were ideal (Fig. 2.3 top). Based on this observation, we propose to use magnetic dipole antennas for millimeter-scale communication systems.

Fig. 2.2 (on the right) shows the prototype design of the millimeter-scale 3D antenna, which can be manufactured using a multi-layer printed circuit board. For various dimensions, the simulated radiation efficiency of the magnetic antenna prototype is provided in Table 2.2 with 1GHz carrier frequency setting. The 'effective' antenna radius in Table 2.2 corresponds to the radius r_{ant} in the theoretical model in Table 2.1 that provides the same radiation efficiency as the 3D magnetic antenna prototype. For the remainder of this chap-

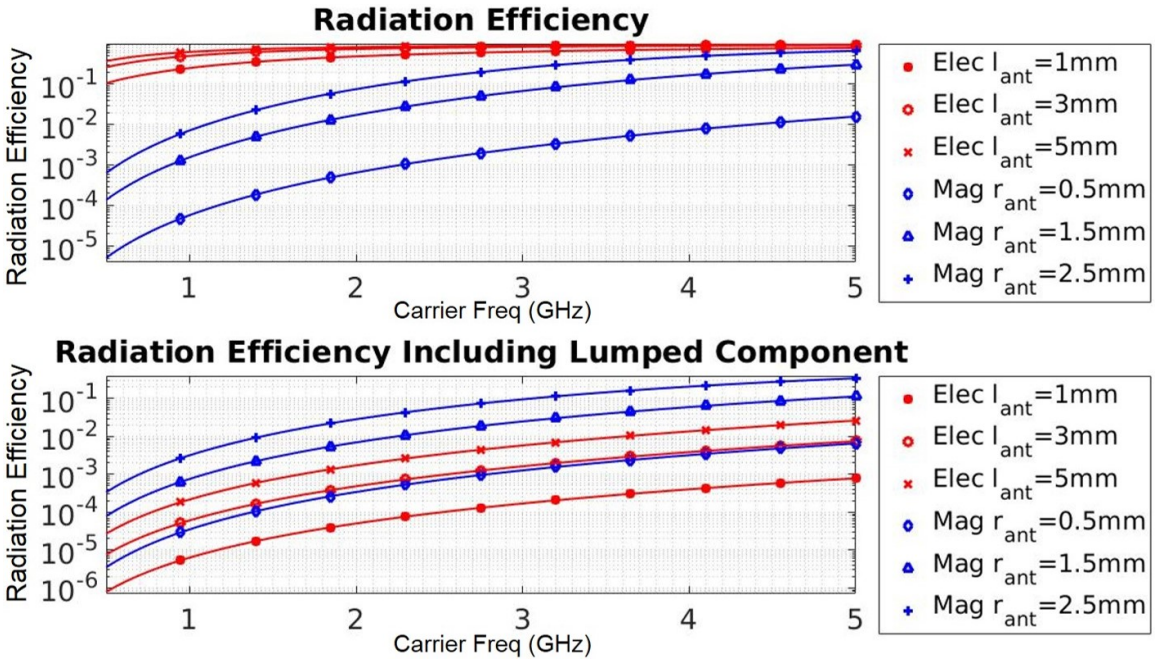


Figure 2.3: Radiation Efficiency of Electric and Magnetic Dipoles

Table 2.2: Simulated 3D magnetic antenna efficiency

Ant. Dimension ($L \times W \times H$ in mm)	Simulated Rad. Eff. $\hat{\eta}_m$ (%)	Effective Radius r_{ant} in mm
$2 \times 2 \times 1$	0.037	0.93
$3 \times 3 \times 2$	0.290	1.46
$4 \times 4 \times 2$	0.527	1.79
$5 \times 5 \times 2$	0.834	2.09

ter, we use the effective antenna radius to represent the dimension of the antenna assuming the efficiency $\hat{\eta}_m$ predicted by Table 2.1.

It is worth noting that although the millimeter-scale magnetic dipole is more efficient than the electric dipole counterpart, its efficiency $\hat{\eta}_m$ is still very poor ($< 1\%$ for $1GHz$ operation as shown in Table 2.2) compared to conventional communication systems with centimeter-scale antennas. The modulation-coding scheme proposed in later sections is carefully designed to address this challenge, and to eventually achieve $> 15m$ distance links for the proposed millimeter-scale system.

2.2.3 Carrier Frequency Selection

The optimal carrier frequency of the proposed system is determined to maximize the signal-to-noise ratio (SNR). Carrier frequency selection affects pathloss of the signal as well as the

antenna efficiency given the millimeter-scale dimension. The RF circuit power efficiency is also affected by the carrier frequency selection.

We consider indoor wireless channel environments including one layer of wall/floor that blocks the line-of-sight (i.e., non-LOS) between the gateway and the sensor node. While a higher carrier frequency is preferred for higher antenna efficiency given a millimeter-scale antenna size (see Fig. 2.3), a lower frequency significantly reduces the pathloss of the signal in line-of-sight as well as the loss from wall/floor penetration. Therefore, the optimal carrier frequency selection needs to strike the balance in this non-trivial tradeoff. For analysis, we use the modified ITU indoor average power propagation loss model [22] given by (2.1).

$$L(f_c, d) = 20log_{10}\left(\frac{4\pi f_c}{c}\right) + 30log_{10}(d) + FL(f_c) - \eta_{sensor} - \eta_{gateway} \ \ [in\ dB] \quad (2.1)$$

In (2.1), c is the speed of light and η_{sensor} (in dB) is the antenna efficiency of the 3D millimeter-scale antenna discussed in Section II.B. The gateway antenna dimension is not our primary concern, and we assume the gateway antenna efficiency $\eta_{gateway}$ is 3dB (half directional antenna). The term $30log_{10}(d)$ dictates the pathloss (using 3.0 exponent instead of 2.0 in the theoretical free-space pathloss) as a function of distance d. $FL(f_c)$ is the additional pathloss due to one layer of floor/wall penetration that is optionally applied to non-LOS scenarios. The original ITU model [22] suggests $FL = 9dB$, $14dB$ and $16dB$ for $915MHz$, $2.4GHz$, and $5.2GHz$, respectively. We modified this term to be linear with carrier frequency as $FL(f_c) = 4 \times (f_c\ in\ GHz) + 7$ in dB based on the measurement results from [23] – [24] (averaged in log domain). For line-of-sight (LOS) scenarios, we make $FL(f_c) = 0$. Fig. 2.4 shows the propagation model (2.1) evaluation in both LOS and non-LOS scenarios for various antenna dimension constraints and carrier frequency selections. It is worth noting that, considering both pathloss and antenna efficiency, $\approx 1GHz$ operation is optimal in the non-LOS scenario when the antenna dimension is limited to $\approx 2mm$. It is because additional pathloss (including wall penetration) penalty offsets the higher antenna efficiency at higher frequencies. Meanwhile, for LOS operation, high frequency ($\geq 5GHz$) operation is desired to minimize the overall propagation loss. The impact of carrier frequency selection on the millimeter-scale communication system energy efficiency, data rate, and link distance will be quantified in Section IV.C.

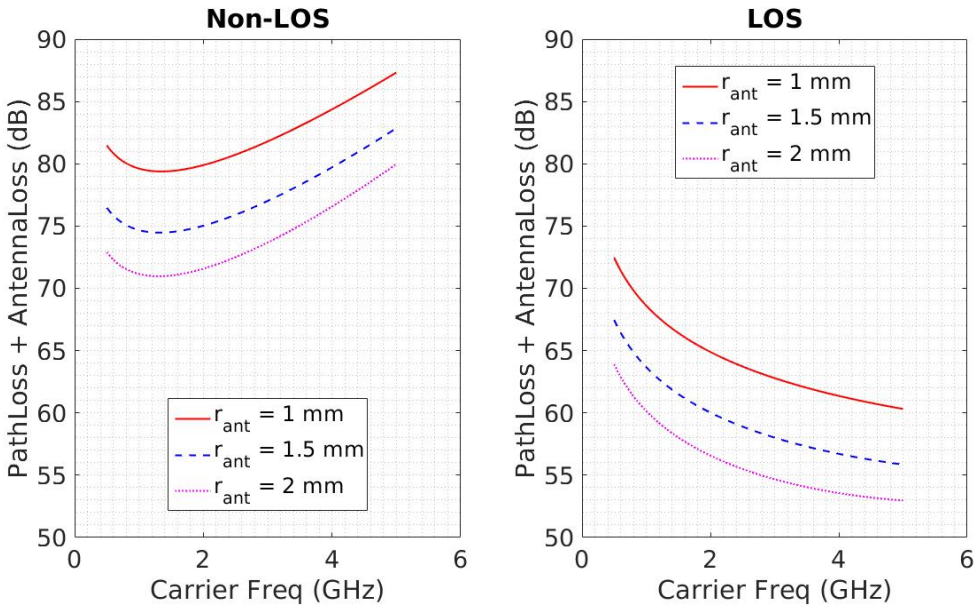


Figure 2.4: Overall propagation loss for NLOS and LOS including pathloss and antenna gain as a function of f_c given distance $d = 1$m

2.2.4 Modulation Scheme

A conventional wireless transmitter uses a phase-locked loop (PLL) for carrier frequency synthesis to support coherent modulation and/or sub-channel tuning for non-coherent modulation. The power consumption of a conventional PLL architecture with the state-of-the-art technology is around $1 - 10mW$ [25] – [26]. This much power overhead is unacceptable for power-constrained millimeter-scale sensor nodes where sustained battery power (and harvested power) is limited to 100s of μW. Furthermore, a PLL typically requires an external crystal oscillator as the phase reference. The commercial crystal has a volume larger than $1.6 \times 1.0 \times 0.5mm^3$ [27], increasing overhead on the overall system dimension.

The proposed system integrates the RF transceiver with a high quality factor (≈ 250) magnetic antenna as described in Section II.B. Replacing a conventional PLL, we propose a 'power oscillator' that injects power into the antenna resonating at a target carrier frequency with a tunable capacitor bank [18][28]. Such implementation has advantages of lower cost, inherent frequency generation, high transmit efficiency ($\approx 30\%$), and inherent antenna matching. A conventional power amplifier is unnecessary for the proposed architecture. The downside of the power oscillator architecture is that it only allows non-coherent modulation schemes.

Note that the transmit efficiency of the proposed power oscillator architecture is maximized at a certain (100s of μA) bias current given $\approx 4V$ power supply [18][28]. This

current consumption is much lower than that of a conventional architecture with a PLL and power amplifier, but it still exceeds the range of the peak battery current of the proposed millimeter-scale system. This implies that transmitting a continuous signal at a reasonable transmit power efficiency is infeasible when the current is directly drawn from the millimeter-scale thin-film battery [8]–[9].

Circumventing this issue, we employ a sparse pulse based non-coherent modulation scheme. In the proposed scheme, the transmit pulse energy on the sensor node is drawn from the energy reservoir capacitor, not directly from the battery. This capacitor is trickle charged by the peak-current limited millimeter-scale thin-film battery while the battery is constantly recharged by the harvested solar energy. This particular power management architecture implies that pulses cannot be repeated until the trickle charging time of the reservoir capacitor, T_{charge}, is elapsed between pulses. This T_{charge} time is regarded as the 'forced idle time' in our modulation scheme.

Exploiting this inherent pulse sparsity, we propose M-ary pulse position modulation (PPM) [29] [30] as the sensor node transmission scheme. In a conventional system, M-ary PPM has been investigated with the purpose of enhancing energy efficiency (a single pulse can contain multiple information bits) [29]. The drawback of M-PPM in a conventional system is its lower bandwidth efficiency as the symbol duration is proportional to M whereas the number of bits per symbol is a function of $log_2(M)$. On the contrary, in our proposed millimeter-scale system, the recharging time of the capacitor is inevitable and usually much longer than the pulse duration. Thus, the symbol duration is dominated by T_{charge} if M is in a reasonable range (≤ 64) as depicted in Fig. 2.5. The forced idle time, T_{charge}, motivates the usage of $M > 2$ to enable higher energy efficiency and to increase the symbol rate at the same time.

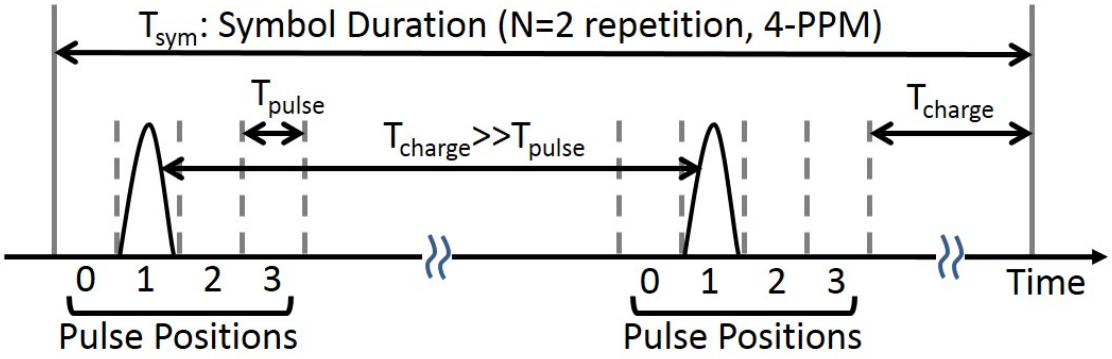


Figure 2.5: N-repetition M-PPM modulation illustration. N=2, 4-PPM example.

The M-PPM symbol error rate is governed by the energy transmitted in a symbol. Since

the proposed system draws energy from the reservoir capacitor, its capacitance limits the maximum energy per pulse constraining the maximum distance. Addressing this issue, we allow N-repetition of pulses to represent a symbol (each pulse is separated by at least T_{charge}, see Fig. 2.5) to expand the link distance beyond the limit of the reservoir capacitance at the potential cost of degraded data throughput.

For the proposed modulation scheme, the bandwidth efficiency penalty of M-PPM over binary-PPM is relatively insignificant because of inherent sparsity. Thus, we utilize numerous possible pulse positions in the sparse transmit signal to absorb convolutional coding redundancy. In Section III and IV, we exploit the sparsity of the modulated signal either 1) to convey multiple information bits per pulse to maximize energy efficiency (and/or data rate) in a short distance link or, 2) to maximize coding gain for longer distance communication. The adaptive N-repetition M-PPM scheme combined with variable rate convolutional coding is a powerful technique to realize an optimized communication link addressing different use-case scenarios and link objectives. Section IV provides in-depth discussion on distance-energy efficiency tradeoffs with the proposed dynamic N-repetition M-PPM modulation scheme with a variable rate convolutional code.

2.2.5 Gateway Guided Synchronization and Link Adaptation

We assume that each ultra-small IoT node is paired with a gateway device such as a smartphone or an access point. Direct communication among ultra-small IoT nodes is not the focus of this work. Since the energy/power efficiency of the gateway is not a primary concern, it is reasonable to assume that the gateway is equipped with powerful signal processing while the sensor node is extremely power constrained. Exploiting this asymmetry leads toward very dissimilar transmitter and receiver configurations for a millimeter-scale sensor node.

The PLL-free crystal-less RF transceiver discussed in Section II.D is a key enabler to achieve ultra-small integration for an energy-autonomous ULP sensor node. However, it imposes a new challenge in frequency predictability. To address this issue, we propose a gateway assisted synchronization protocol that is initiated by sensor node transmission. In the proposed protocol, the gateway estimates the RF carrier frequency offset (CFO) and the baseband sampling frequency offset (SFO) between the gateway and the sensor node via multi-hypotheses correlations. Once the gateway identifies CFO and SFO, it sends a customized packet that 'pre-compensates' carrier and baseband frequency offsets for a particular sensor node. This gateway-assisted synchronization allows PLL-free crystal-less sensor node implementation, enabling ultra-low power and ultra-small system integration.

It also eliminates the need for timing and frequency synchronization at the millimeter-scale sensor node, which is often the most power-dominant processing in conventional wireless receiver baseband.

The proposed protocol works in the following procedure (illustrated in Fig. 2.6):

1) The sensor node initiates the communication by sending a set of sparse pulses, Synch_HDR, with a pre-defined (node ID dependent) pseudo-random interval.

2) Gateway is always listening and it detects the Synch_HDR via the multi-hypotheses correlation (algorithm in Section III.B) covering all possible ranges of the CFO and SFO. The instantaneous SNR is obtained based on the peak correlation value. The CFO and SFO are estimated at the gateway as a result of Synch_HDR detection.

3) After transmitting the Synch_HDR, the sensor node enters the receive mode. Demodulation at the sensor node starts at a predefined delay (turnaround time, T_{turn}), calculated using the sensor node baseband clock (implemented without PLL and crystal reference; see [31] for an example). At the sensor node, the correlation process searching for the symbol boundary is unnecessary if the symbol from the gateway arrives at the precise timing.

4) Gateway calculates the turnaround time using the estimated SFO to synchronize with the sensor node. It also adjusts carrier frequency f_c, based on the CFO estimation. In parallel, gateway solves the link adaptation problem (Section IV) to identify modulation parameters for the sensor node to maximize an objective function (data rate or energy efficiency) given the instantaneous SNR. Gateway sends the message to the sensor node at the exact symbol boundary that the sensor node is expecting. The message contains a command to select the optimal operating mode for the sensor node.

5) The sensor node sends more messages using the optimal mode dictated by the gateway.

Dynamic modulation parameters such as the pulse width, coding rate, and modulation size (M) for M-PPM allow the proposed system exploiting the rich tradeoff space to maximize a specific objective function such as the data rate and/or link distance. The mathematical formulation and analysis of the link adaptation will be discussed in Section IV.

In this work, we assume the sensor node is the main source of the wireless communication data traffic. Typical messages from the gateway to the sensor node are assumed to be short commands that are less bandwidth demanding. This is a valid assumption for majority of IoT applications where distributed sensor nodes collect various sensing data including audio/image.

Therefore, our link adaptation strategy with dynamic sparse M-PPM modulation is only

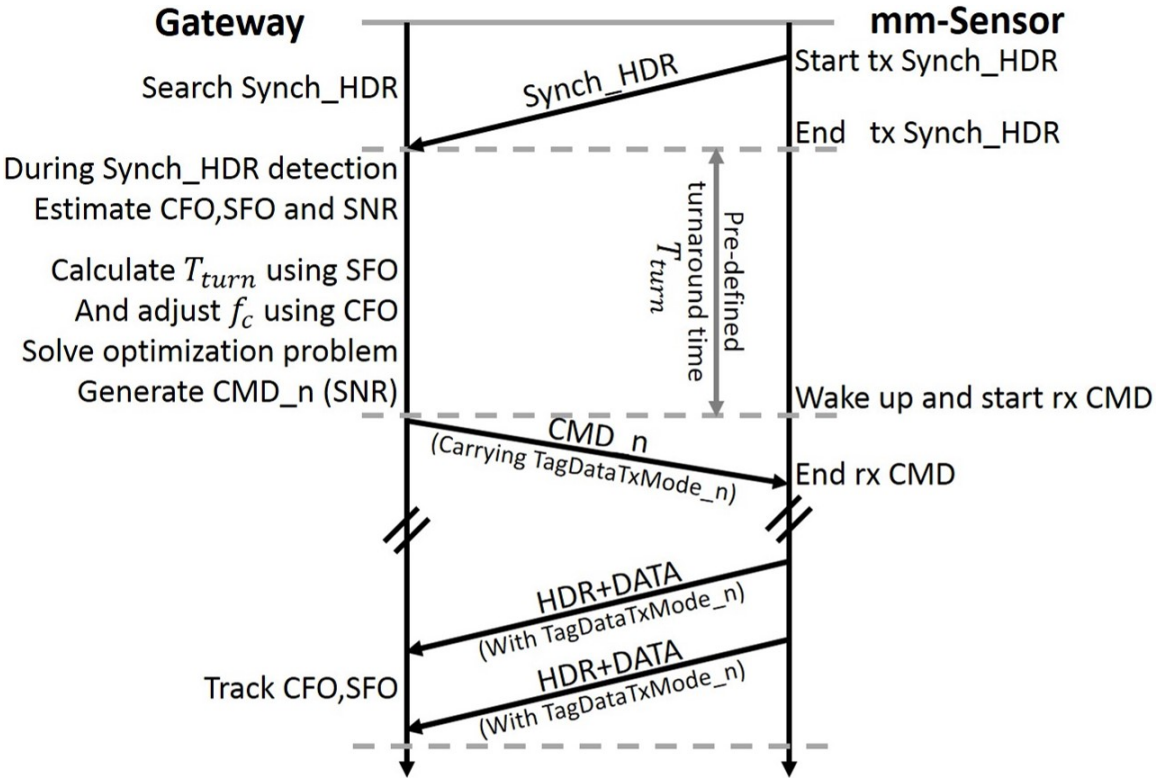


Figure 2.6: Gateway guided synchronization and link adaptation

applied to sensor node transmission. The gateway transmit signal, on the other hand, uses a simple static modulation scheme such as on-off-keying (OOK). Note that the gateway transmit signal does not have to be sparse. In the proposed scheme, the gateway adjusts its transmit power so that a certain minimum signal sensitivity level at the sensor node receiver operating with 10s of μW power budget (sustainable with thin-film battery power) is always satisfied. The comprehensive survey on the state-of-the-art ultra-low power ($< 100\mu W$) OOK receiver designs [32] suggests that this scheme is certainly feasible especially when the baseband processing is greatly simplified by the proposed gateway-guided synchronization. Thus, our proposed scheme focuses on the sensor node transmission, assuming the other direction (gateway transmission - sensor node reception) does not limit the system performance.

2.3 System Modeling

2.3.1 Sensor Node Transmission Signal Modeling

The proposed modulation and coding scheme for sensor node transmission is depicted in Fig. 2.7. The information bit stream is fed into a multi-rate convolutional code encoder, then mapped into M-PPM pulse signals. The M-PPM pulse is optionally repeated N times, and pulse shaping is performed with tunable pulse width (T_{pulse}).

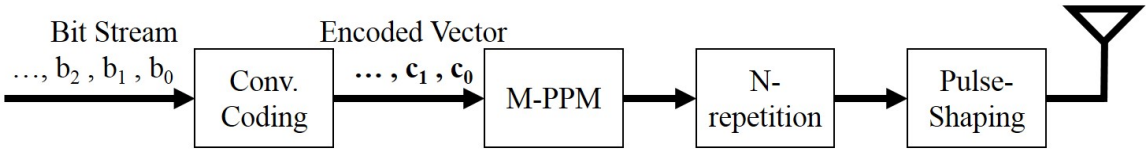


Figure 2.7: Baseband Processing at Sensor Node

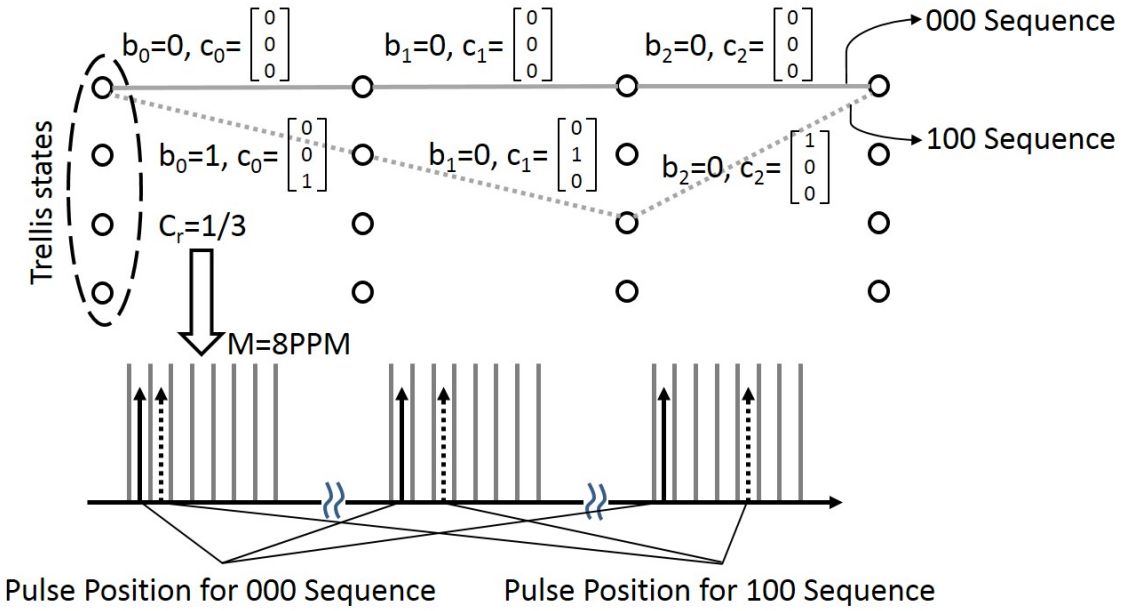


Figure 2.8: $\nu = 3$, $C_r = 1/3$ coding example. M-PPM size is tightly coupled with C_r. $M = 2^{1/C_r} = 8$ PPM is used.

In the proposed modulation and coding scheme, the M-PPM modulation is tightly coupled with convolutional encoding. The modulation-coding rate (C_r) is dynamically chosen from the set; $C_r \in \{..., 3, 2, 1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, ...\}$. Convolutional encoding is bypassed when $C_r \geq 1$, and higher data rate is achieved by using $M = 2^{Cr}$ to carry C_r information bits per symbol. On the other hand, when convolutional encoding is enabled ($C_r < 1$), M is

chosen to be $2^{1/C_r}$ to convey $1/C_r$ coded bits and, equivalently, one information bit per M-PPM symbol. M and C_r satisfy the relationship (2.2). In fact, the usage of $C_r < \frac{1}{2}$ coding rate is motivated by the sparsity of the transmit signal, where numerous pulse positions are available without a significant data rate penalty.

$$M = \begin{cases} 2^{C_r} & C_r \geq 1 \\ 2^{\frac{1}{C_r}} & C_r < 1 \end{cases} \tag{2.2}$$

$$\mathbf{c_j} = \begin{cases} [b_{jC_r} \ \dots, \ b_{(j+1)C_r-1}]^T & C_r \geq 1 \\ G(\sigma_j, b_j, C_r) & C_r < 1 \end{cases} \tag{2.3}$$

We denote the j-th output from the encoder as $\mathbf{c_j}$, a vector of size $log_2 M \times 1$, given input bit stream $(b_0, b_1, ...)$. $G(\sigma_j, b_j, C_r)$ is the convolutional code generator function that produces an $1/C_r \times 1$ output vector per single input bit b_j given the convolutional code trellis state σ_j. The trellis state is updated by $\sigma_j = \sum_{l=1}^{\nu-1} b_{j-l} 2^{l-1}$ where ν is the code constraint length. We use convolutional code generator functions specified in [33] for various coding rate C_r and ν. Note that each $\mathbf{c_j}$ vector is mapped to a single M-PPM symbol regardless of C_r. The mapping between the M-PPM symbol index $m_j \in \{0, 1, ..., M-1\}$ and $\mathbf{c_j}$ is given by $m_j = \mathbf{c_j}^T \mathbf{p}$ where $\mathbf{p} = [2^{\log_2 M-1}, ..., 2^1, 2^0]^T$ is the M-PPM position mapping vector. Fig. 2.8 shows an example of $\nu = 3$ convolutional coding with $C_r = 1/3$, $M = 2^{1/C_r} = 8$ PPM.

When the energy per pulse is limited by the capacity of the energy reservoir, we increase the symbol energy by N repetition of M-PPM pulses. The n-th repeated pulse position of the j-th symbol is denoted by $\tau_p(j, n)$ (2.4).

$$\tau_p(j, n) = T_{\text{pulse}} m_j + (j-1) T_{\text{sym}} + (n-1) T_{\text{idle}}, \quad n = 1, ..., N \tag{2.4}$$

In (2.4), T_{sym} is the symbol duration that consists of N pulses. T_{pulse} is the pulse width, which can be dynamically configured as a result of link adaptation discussed in Section IV. $T_{\text{idle}} = max\{T_{\text{charge}}, \ MT_{\text{pulse}}\}$ is the forced idle time between pulse repetition determined by the maximum of the energy reservoir capacitor charging time T_{charge} and the non-overlapping spacing for M-PPM MT_{pulse}. In case the system's thin-film battery current is very limited, the charging time dominates $T_{\text{idle}} = T_{\text{charge}} \gg MT_{\text{pulse}}$ for a reasonable M (≤ 64). The symbol duration of an N repetition M-PPM symbol is obtained by (2.5). Given a pulse shape function $p(t)$ with support [0,T_{pulse}], the sensor node transmit signal is represented by (2.6), where $*$ stands for convolution and $\delta(x)$ is the Dirac-Delta function.

$$T_{\text{sym}} = MT_{\text{pulse}} + (N-1) T_{\text{idle}} + T_{\text{charge}} \tag{2.5}$$

$$s(t) = p(t) * \sum_{j} \sum_{n=1}^{N} \delta(t - \tau_p(j,n)) \tag{2.6}$$

2.3.2 Synchronization at Gateway

Waiting for Synch_HDR that initiates the communication, the gateway is always listening. The proposed Synch_HDR detection process at the gateway provides reliable timing and frequency offset synchronization, which is critical to enable PLL-less crystal-free implementation for the millimeter-scale sensor node. The gateway employs non-coherent demodulation and synchronization that is based on received sample power. Coherent demodulation is infeasible because the sensor node cannot maintain phase coherency without a PLL.

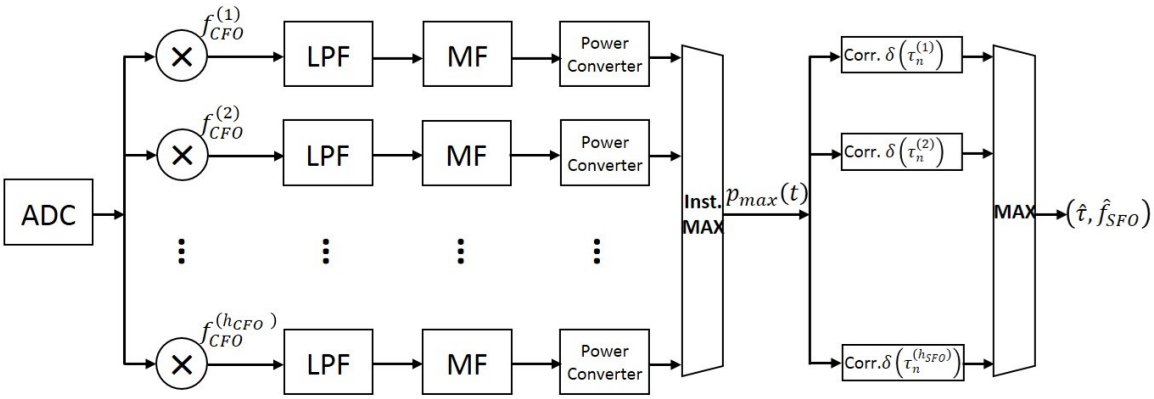


Figure 2.9: Synchronization datapath at the gateway for carrier frequency offset and sampling frequency offset estimation

Fig. 2.9 depicts the Synch_HDR detection process at the gateway with carrier frequency offset (CFO) and sampling frequency offset (SFO) estimation. The h_{CFO} and h_{SFO} are the number of hypotheses for discretized CFO and SFO respectively. During Synch_HDR detection, the incoming baseband ADC samples are mixed with various CFO hypotheses ($f_{CFO}^{(1)}, f_{CFO}^{(2)}, ..., f_{CFO}^{(h_{CFO})}$). Each CFO mixer output is low-pass filtered, power converted, and then convoluted with a matched filter (MF). The instantaneous maximum output, $p_{max}(t)$ is selected among h_{CFO} MF outputs until the Synch_HDR is detected. Each maximum ouput is associated with a specific CFO hypothesis, $f_{CFO}^{max}(t)$, at time instance t.

This instantaneous maximum output signal $p_{max}(t)$ is then correlated with h_{SFO} impulse sequences. The j-th correlation sequence $\sum_{n=1}^{N_p} \delta(t - \tau_n^{(j)})$ has pulse positions $\tau_n^{(j)}$, $n = 1, 2, ..., N_p$ that are determined by the predefined pulse interval in the Synch_HDR that is adjusted according to the j-th SFO hypothesis. N_p is the number of pulses in a

Synch_HDR. The Synch_HDR is successfully detected when the maximum from multiple hypotheses correlations exceeds a certain threshold. Consequently, the SFO estimate $\hat{f}_{SFO}$ is obtained by a particular SFO hypothesis that maximizes the correlation as in (2.7). In addition, the CFO estimate $\hat{f}_{CFO}$ is computed by (2.7), taking the average of $f_{CFO}^{max}(t)$ sampled at the pulse positions given by SFO estimation.

$$\hat{f}_{SFO} = \underset{j=1,\ldots,h_{SFO}}{\operatorname{argmax}} \int \sum_{n-1}^{N_p} p_{max}(t)\delta(t+\tau_n^{(j)})dt, \quad \hat{f}_{CFO} = \frac{1}{N_p}\sum_{n=1}^{N_p} f_{CFO}^{max}(\tau_n^{(*)}) \tag{2.7}$$

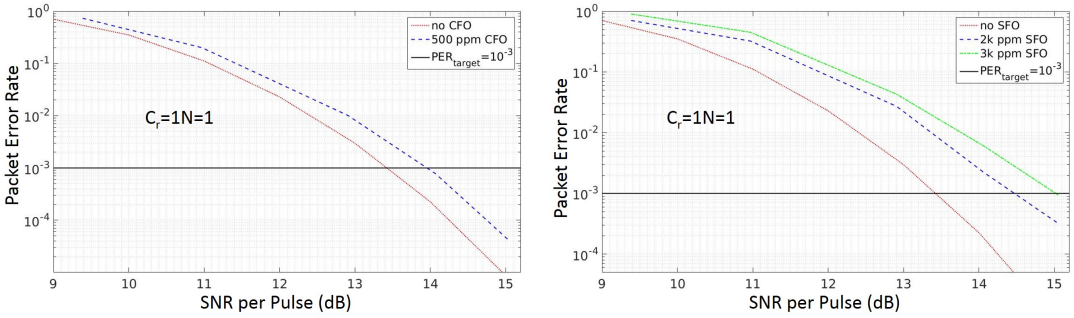


Figure 2.10: Performance of CFO estimation (left) and SFO estimation (right)

The left plot in Fig. 2.10 shows the performance of this CFO estimation scheme. 0.6dB performance degradation is observed from the ideal (no CFO) case when CFO is set to $500ppm$ ($= 500kHz$ at $f_c = 1GHz$, simulated with $C_r = 1$, and $N = 1$). Once the Synch_HDR is detected, the mixing path that corresponds to $\hat{f}_{CFO}$ remains active while all other mixers are disabled. The matched filter output $p_{max}(t)$ is resampled using the SFO estimate $\hat{f}_{SFO}$.

After successful Synch_HDR detection, multiple hypotheses correlations to evaluate (2.7) remain active during the data modulation process tracking the residual SFO that might affect the system performance. Since $p_{max}(t)$ is resampled with $\hat{f}_{SFO}$, the SFO hypotheses ($f_{SFO}^{(1)}, f_{SFO}^{(2)}, \ldots, f_{SFO}^{(h_{SFO})}$) can now be readjusted with finer granularity. As data pulse demodulation continues, expected pulse positions $\tau_n^{(j)}$ that originally all belong to Synch_HDR are sequentially replaced by detected data pulse positions for the residual offset tracking in a decision feedback fashion. The oldest pulse position in the hypothesis is replaced by the latest detected pulse position, and the $\hat{f}_{SFO}$ tracking continues until the end of the packet.

The right plot in Fig. 2.10 shows the simulation performance of the proposed SFO estimation and tracking algorithm. The simulation results confirm that performance degradation due to SFO is limited to an acceptable range ($<$ 1dB SNR loss) when $\leq 2000ppm$

(0.2%) SFO is tested. This SFO requirement is very reasonable for ultra-low power ($< 1\mu W$) clock design [31] that does not require a reference crystal. Based on the synchronization algorithm performance shown in Fig. 2.10, we argue that the proposed scheme mitigates CFO and SFO well, and the impact of residual CFO and SFO is insignificant. Hence, we will assume perfect gateway guided synchronization for the remaining sections of the chapter.

2.3.3 Demodulation Performance Modeling

In this section, we derive analytical packet error rate performance expressions for the proposed modulation-coding scheme.

2.3.3.1 Uncoded ($C_r \geq 1$) cases

We employ a non-coherent energy detector at the receiver (i.e., gateway) to demodulate the N-repetition M-PPM signal transmitted from the sensor node. A channel with complex additive white Gaussian noise $\mathcal{CN}(0, N_0)$ is assumed throughout the performance analysis. An $N \times 1$ vector $\mathbf{r}$ denotes the set of matched filter outputs sampled at the correct N pulse positions for a N-repetition M-PPM symbol. Similarly, let $\mathbf{e}$ be an $N \times 1$ vector, the set of matched filter outputs sampled at incorrect symbol pulse positions. The pulse energy is normalized to one without loss of generality throughout the analysis. Therefore, the symbol is correctly detected when $\|\mathbf{r}\|^2 > \|\mathbf{e}\|^2$, for all $\mathbf{e}$'s that correspond to $M-1$ possible error positions. Assuming all symbols are equally probable, the probability of correct symbol detection P_c is given by (2.8) where $P\{\}$ denotes probability. Note that $X_r = \frac{\|\mathbf{r}\|^2}{N_0/2}$ is a non-centralized chi-square distributed random variable with a degree of freedom $2N$ and non-centralized parameter $s = \frac{N}{N_0/2}$. $X_e = \frac{\|\mathbf{e}\|^2}{N_0/2}$ is centralized chi-square distributed with a degree of freedom $2N$.

$$P_c(N, M, N_0) = \left(P\left\{X_r > X_e\right\}\right)^{M-1} \tag{2.8}$$

Since X_e has an even degree of freedom, it has a closed-form expression cdf [34]. The analytical expression of X_r's pdf is available in [30]. Therefore, P_c can be rewritten as (2.9) where $SNR = \frac{1}{N_0}$, $s = 2N \cdot SNR$ and I_{N-1} is the modified Bessel function of the first kind.

$$P_c(N, M, SNR) = \int_0^\infty \left(1 - e^{-\frac{x}{2}} \sum_{j=0}^{N-1} \frac{1}{j!} \left(\frac{x}{2}\right)^j\right)^{M-1} \frac{1}{2} \left(\frac{x}{s^2}\right)^{\frac{N-1}{2}} e^{-\frac{s^2+x}{2}} I_{N-1}(s\sqrt{x}) dx \tag{2.9}$$

When convolutional encoding is unused ($C_r \geq 1$), the number of symbols in a packet containing L information bits is L/C_r. Using (2.9), the PER is obtained by (2.10).

$$PER = 1 - P_c(N, M, SNR)^{L/C_r} \tag{2.10}$$

2.3.3.2 Convolution Coded ($C_r < 1$) Cases

When convolutional coding is enabled ($C_r < 1$), the non-coherent energy detection is performed along with the optimal maximum likelihood sequence estimation (MLSE) [34] at the gateway. Unlike a conventional soft-input Viterbi decoding where the log likelihood ratio is used as the branch metric [35], the matched filter output power sampled at the expected pulse position is directly used as the branch metric in our scheme. The likelihood of each symbol sequence is represented by accumulated branch metric (i.e., the integrated pulse energy) along the trellis transition path. At each trellis state of MLSE, the branch with the maximum accumulated branch metric (or the maximum integrated pulse energy) is selected updating the accumulated metric for each state.

To arrive at an analytical expression of the convolution coded packet error rate, we use the union bound (2.11), which provides a strict but tight upper bound of the actual PER [34]. The trellis length L of MLSE is equal to number of information bits (convolutional code input) in our modulation-coding scheme when $C_r < 1$. Without loss of generality, we assume the correct MLSE sequence corresponds to the all zero input sequence for our PER analysis.

$$PER \leq 1 - \left(\prod_{l=1}^{L} \prod_{k=1}^{l} \left(1 - P_{e,\text{pair}}(l, k, N)\right)^{A(l,k)} \right)^{L} = 1 - \left(\prod_{l=1}^{L} \prod_{k=1}^{l} \left(1 - P_c\left(Nk, 1/C_r, N_0\right)\right)^{A(l,k)} \right)^{L} \tag{2.11}$$

In (2.11), $A(l, k)$ is the number of 'length-l distance-k simple error' events that diverge from the all-zero sequence from the beginning of the trellis and merge (for the first time) to the all-zero sequence after l branch transitions. A length-l, distance-k simple error event has k different pulse positions from the all-zero sequence over l trellis transitions. $P_{e,\text{pair}}(l, k, N)$ is the probability of the pairwise error event, which occurs when the all-zero sequence has a less accumulated branch metric than a length-l distance-k simple error given N repetition modulation and noise power of N_0. In fact, it is straightforward to show that $P_{e,\text{pair}}(l, k, N) = P_c(Nk, 1/C_r, N_0)$ using (2.9) when $M = 2^{1/C_r}$ and $C_r \in \{\frac{1}{2}, \frac{1}{3}, ...\}$. Note that $A(l, k)$ is dictated by the convolutional code generator function $G(\sigma_j, b_j, C_r)$ as well as the M-PPM encoding. We empirically evaluate $A(l, k)$ for all convolutional codes considered in this work.

Fig. 2.11 shows side-by-side comparisons between simulated PER and analysis results given by (2.9) and (2.11) for various uncoded ($C_r \geq 1$) and coded ($C_r < 1$) cases. The packet length L is 128 bits for all cases. The average pulse SNR (on x-axis of Fig. 2.11) is $1/N_0$ assuming normalized pulse energy. The uncoded PER analysis exactly matches with the simulation (with triangle dot), while the union bound of the coded PER (2.11) is proven to be tight for all coded cases (simulation results with circle dot). Hence, for the remaining sections, we use the union bound (2.11) with equality to represent the PER when convolutional coding is enabled. As Fig. 2.11 shows, the proposed modulation-coding scheme enables the system operating at low SNRs ($\approx 0dB$ per pulse) when $C_r \leq 1/2$ convolutional coding is combined with $N \geq 1$ repetition.

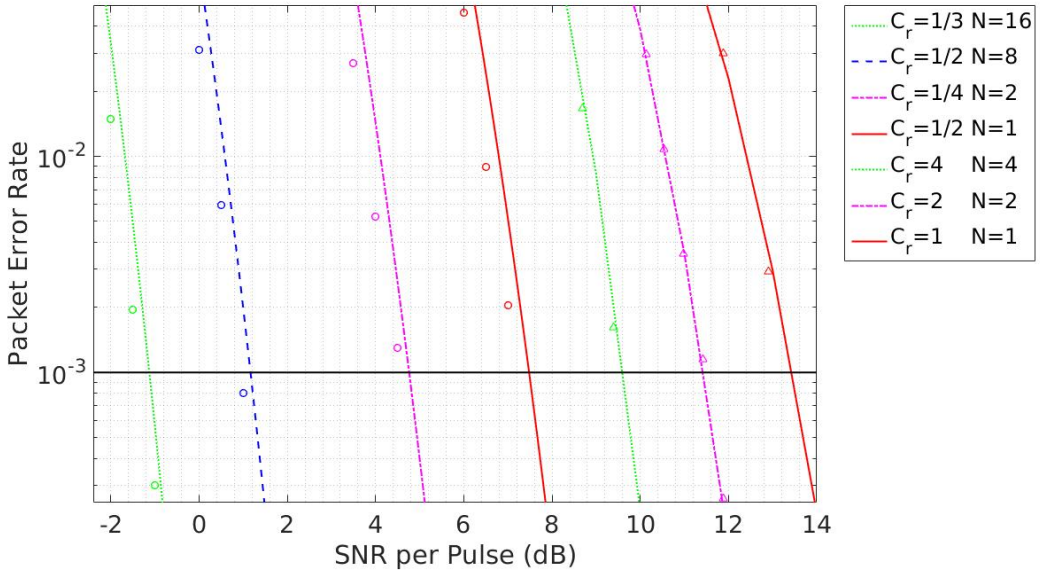


Figure 2.11: Performance of modulation-coding scheme. Analysis for uncoded cases. Tight union bound for coded cases.

2.3.4 Data Rate and Energy Efficiency Modeling

The data rate of the proposed system is defined by the number information bits transmitted per unit time. The proposed system supports a wide range of data rates by changing modulation-coding parameters; N, C_r, and T_{pulse} dynamically. The number of information bits contained in a symbol is $\lceil C_r \rceil$. That is, a single symbol conveys a single information bit when the convolutional coding is enabled ($C_r < 1$). Without error correction coding, on the other hand, $C_r (\geq 1)$ bits are transmitted per symbol using $M = 2^{C_r}$ PPM. Using (2.5) for the symbol duration, the system data rate R is obtained by (2.12). Recall that

$T_{\text{idle}} = max(MT_{\text{pulse}},\ T_{\text{charge}})$ and T_{charge} is a function of T_{pulse} given battery current limitation. M is a function of C_r as given in (2.2). Therefore, the data rate R of the system is fully determined by three modulation-coding parameters; N, C_r, and T_{pulse}.

$$R = \frac{\lceil C_r \rceil}{T_{\text{sym}}} = \frac{\lceil C_r \rceil}{MT_{\text{pulse}} + (N-1)T_{\text{idle}} + T_{\text{charge}}} \tag{2.12}$$

Achieving the maximum energy efficiency is one of the primary objectives of the proposed link adaptation system. The energy per information bit for millimeter-scale sensor node transmission has the expression (2.13).

$$E_b = \frac{\text{Energy per symbol}}{\text{Number of info bits per symbol}} = \frac{P_{ckt} N\ T_{\text{pulse}}}{\lceil C_r \rceil} = \frac{P_{TX} N\ T_{\text{pulse}}}{\eta_{ckt} \lceil C_r \rceil} \tag{2.13}$$

In (2.13), P_{ckt} is the constant power consumption of the 'power oscillator' circuit proposed in section II.D. Recall that the efficiency of the circuit is maximized at a certain constant bias condition. We assume the efficiency $\eta_{ckt} = 0.15$ is achieved when $P_{ckt} = 3.5mW$ using an architecture similar to [28]. While a constant transmit power level $P_{TX} = \eta_{ckt} P_{ckt}$ is delivered to the antenna maintaining the maximum circuit efficiency, we adjust T_{pulse}(i.e., signal bandwidth) and/or C_r for link adaptation in various SNR conditions. Note that the transmitter consumes near zero power during the idle time (T_{charge} and T_{idle}) between pulses when only the ULP oscillator [31] is active to control the N-repetition M-PPM pulse timing.

2.4 System Optimization and Link Adaptation

In this section, we will formulate link adaptation optimization problems for the proposed energy-aware ultra-small IoT communication system. We first introduce system constraints of the millimeter-scale sensor node, and then formulate formal link adaptation optimization problems for 1) the maximum link distance, 2) the maximum data rate, and 3) the maximum energy efficiency objective functions.

The design parameters and system constants for a realistic millimeter-scale sensor node communication system are specified in Table 2.3. These constants are used throughout the system link adaptation study, unless specified otherwise. The proposed system employs three modulation-coding parameters that can be dynamically adjusted for link adaptation; C_r, N, and T_{pulse}. These are discrete variables as shown in Table 2.3. All other parameters such as T_{charge}, T_{idle}, E_b are all implicitly specified by the selection of C_r, N, and T_{pulse}.

Since the pulse energy is drawn from the capacitor, the energy per pulse is limited by the energy stored in the reservoir capacitor C_{res}. The upper bound on the pulse width, T_{pulse},

Table 2.3: System Design Parameters, Constants and Adaptive Modulation-Coding Parameters

System Design Parameters	**Values**
Antenna Dimension, r_{ant}	$1.5mm$
Carrier Frequency, f_c	$1GHz$
Reservoir Capacitor, C_{res}	$150nF$
Battery Current, I_{bat}	$30\mu A$
Circuit Operating Condition	$V_{min} = 2.6V$, $V_{DD} = 3.6V$, $P_{ckt} = 3.5mW$, $\eta_{ckt} = 0.15$
Target PER, PER_{target} (Packet Length=128bit)	10^{-3}
Target Data Rate, R_{target}	$10k$bit/s
Adaptive Modulation-Coding Parameters	**Values**
Coding Rate C_r	$\left\{\frac{1}{6}, \frac{1}{5}, \frac{1}{4}, ..., 1, 2, ..., 8\right\}$
Pulse Repetition N	$\{1, 2, ..., 16\}$
Pulse Width T_{pulse} in μs	$\{0.05,\ 0.1,\ 0.2,\ 0.4,\ ...,\ 51.2,\ 102.4, ...\}$

is obtained by (2.14) where V_{min} is the minimum voltage required for transmitter circuit functionality. The recharging time T_{charge} (2.15) is required between pulses to restore charge in the reservoir capacitor.

$$T_{\text{pulse}} \leq \frac{\text{Energy in reservoir cap}}{2P_{ckt}} = \frac{C_{\text{res}}({V_{DD}}^2 - {V_{min}}^2)}{2P_{ckt}} \tag{2.14}$$

$$T_{\text{charge}}(C_{\text{res}}, T_{\text{pulse}}) = \frac{C_{\text{res}}(V_{DD} - \sqrt{{V_{DD}}^2 - \frac{2P_{ckt}T_{\text{pulse}}}{C_{\text{res}}}})}{I_{bat}} \tag{2.15}$$

In Section III.C, the packet error rate of the proposed system was analyzed as a function of the pulse SNR. Given a transmit pulse width T_{pulse}, the pulse SNR at the gateway can be obtained by (2.16), where NF_{gateway} is the noise figure of the gateway receiver.

$$SNR(T_{\text{pulse}}, d) = 10log_{10}(P_{TX}N\,T_{\text{pulse}}) - N_0 - L(d, f_c) - NF_{\text{gateway}}\ [in\ dB] \tag{2.16}$$

The inverse function of the PER expression (2.10) and (2.11) is difficult to obtain. Therefore, to satisfy a certain PER performance requirement PER_{target}, we numerically evaluate PER expressions (2.10)(2.11) and identify the target SNR, $SNR_{\text{target}}(N, T_{\text{pulse}}, C_r)$ as a function of N, T_{pulse} and C_r, given PER_{target}. This mapping can be obtained off-line and stored in the gateway memory for real-time link adaptation. For all feasible link adaptation solutions, $SNR(T_{\text{pulse}}, d) \geq SNR_{\text{target}}(N, T_{\text{pulse}}, C_r)$ has to be satisfied.

The constraint for the target data rate, R_{target} shall be given as (2.17). Note that M and

T_{idle} are determined by C_r and T_{pulse}, respectively.

$$R(N, T_{\text{pulse}}, C_r) = \frac{\lceil C_r \rceil}{M\ T_{\text{pulse}} + (N-1)T_{\text{idle}} + T_{\text{charge}}} \geq R_{\text{target}} \tag{2.17}$$

In the proposed system, the Synch_HDR (in Fig. 2.9) that initiates the communication between the sensor node and the gateway is designed for the worst case distance. Once the gateway receives this Synch_HDR, it analyzes the correlation output and estimates the channel state information such as the link distance d to evaluate the instantaneous SNR (2.16). The gateway then solves the link adaptation problem for a given objective function, and notifies the sensor node the optimum mode selection result.

2.4.1 Maximum Distance Objective

The maximum link distance between the millimeter-scale sensor node and the gateway is obtained by solving the optimization problem (2.18) for dynamic modulation-coding parameters N, T_{pulse} and C_r. Quantifying the solution of this link adaptation problem is essential to verify system feasibility for a target application scenario.

$$\begin{aligned}
&\text{Maximize:} && d \\
&\text{Subject to:} && T_{pulse} \leq \frac{C_{\text{res}}({V_{DD}}^2 - {V_{min}}^2)}{2P_{ckt}} \\
& && T_c(C_{\text{res}}, T_{\text{pulse}}) = \frac{C_{\text{res}}\left(V_{DD} - \sqrt{V_{DD}^2 - \frac{2P_{ckt}T_{\text{pulse}}}{C_{\text{res}}}}\right)}{I_{bat}} \\
& && \frac{\lceil C_r \rceil}{M\ T_{\text{pulse}} + (N-1)T_{\text{idle}} + T_{\text{charge}}} \geq R_{\text{target}} \\
& && SNR(T_{\text{pulse}}, d) \geq SNR_{\text{target}}(N, T_{\text{pulse}}, C_r)
\end{aligned} \tag{2.18}$$

Fig. 2.12 shows the optimization result for the maximum link distance when the target minimum data rate R_{target} ranges from 10^2 to 10^6 bit/s. The target PER is set to 10^{-3} with packet length of 128 bits. In Fig. 2.12 on the left, x-axis is the variable target data rate and y-axis is the maximum link distance attainable by operating the system at the optimal N, T_{pulse}, and C_r. The result is shown for the NLOS where one layer of wall penetration loss is considered using (2.1) in computing SNR. The solid black line on Fig. 2.12 left is the distance that can be supported by a static scheme using $N = 1$, $C_r = 1$ (binary PPM without coding), and T_{pulse} = $1\mu s$. The data rate of this static scheme is fixed to 29 $k$bit/s while it can operate up to $3.4m$ distance in the NLOS scenario. For a slightly higher (31 $k$bit/s) data rate, the optimization result indicates 60% ($2m$) distance gain over

this particular static scheme by operating with $N = 1$, $C_r = 1/3$, and $T_{\text{pulse}} = 0.8\mu s$. In this data point ($R_{\text{target}} = 30k$bit/s), the optimal $C_r = 1/3$ is lower than the static scheme coding rate ($C_r = 1$) but data rate degradation is avoided using a shorter pulse width (thus shorter T_{charge}), while the longer distance is achieved by the error correction coding. The optimal link adaptation result demonstrates graceful tradeoffs in link distance vs. data rate as Fig. 2.12 (left) shows. For the proposed millimeter-scale system, the optimal distance ranges from $1m$ to $> 30m$ when the target data rate is set to 10^6 and 10^2 respectively in the NLOS scenario.

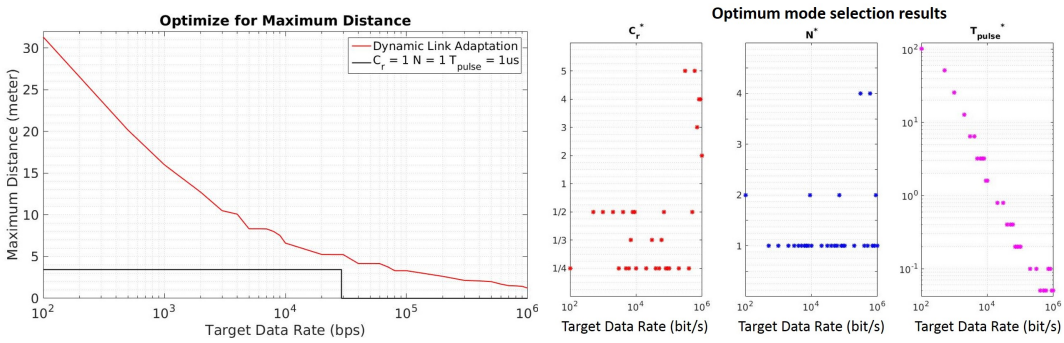


Figure 2.12: Maximum distance objective link adaptation result

Fig. 2.12 (right) depicts the optimal mode $(N, C_r, T_{\text{pulse}})$ selection results for the maximum distance objective link adaptation. Note that we consider discretized $T_{\text{pulse}} \in \{0.05,\ 0.1,\ 0.2, ...\}\mu s$ to render more realistic hardware implementation. When the target data rate is low, longer pulse widths and smaller coding rates are preferred to maximize the distance. But it is worth noting non-monotonic behavior in selection of C_r when the system is allowed to adjust N and T_{pulse} optimally. When the target data rate is high ($> 500k$bit/s), uncoded ($C_r \geq 1$) M-PPM modulation is selected as the optimal. To maximize the link distance, the optimal system often selects $N > 1$ in addition to convolutional coding, while the optimal T_{pulse} monotonically decreases in general to meet a higher data rate target.

2.4.2 Maximum Data Rate and Energy Efficiency Objectives

The link adaptation problem for the maximum data rate is formulated as (2.19), where the objective is to maximize the data rate R given a link distance target d_{target}.

$$\begin{aligned}
\text{Maximize:} \quad & R = \frac{\lceil C_r \rceil}{M\, T_{\text{pulse}} + (N-1)T_{\text{idle}} + T_{\text{charge}}} \\
\text{Subject to:} \quad & T_{\text{pulse}} \leq \frac{C_{\text{res}}({V_{DD}}^2 - {V_{min}}^2)}{2P_{ckt}} \\
& SNR(T_{\text{pulse}}, d_{\text{target}}) \geq SNR_{\text{target}}(N, T_{\text{pulse}}, C_r) \\
& T_c(C_{\text{res}}, T_{\text{pulse}}) = \frac{C_{\text{res}}\left(V_{DD} - \sqrt{V_{DD}^2 - \frac{2P_{ckt}T_{\text{pulse}}}{C_{\text{res}}}}\right)}{I_{bat}}
\end{aligned} \tag{2.19}$$

Similarly, the link adaptation for the maximum energy efficiency has the form of (2.20), where the energy per information bit is minimized for a given link distance target d_{target} satisfying the minimum data rate constraint R_{target}.

$$\begin{aligned}
\text{Minimize:} \quad & E_b = \frac{P_{TX} N\, T_{\text{pulse}}}{\eta_{ckt}\lceil C_r \rceil} \\
\text{Subject to:} \quad & T_{pulse} \leq \frac{C_{\text{res}}({V_{DD}}^2 - {V_{min}}^2)}{2P_{ckt}} \\
& T_c(C_{\text{res}}, T_{\text{pulse}}) = \frac{C_{\text{res}}\left(V_{DD} - \sqrt{V_{DD}^2 - \frac{2P_{ckt}T_{\text{pulse}}}{C_{\text{res}}}}\right)}{I_{bat}} \\
& \frac{\lceil C_r \rceil}{M\, T_{\text{pulse}} + (N-1)T_{\text{idle}} + T_{\text{charge}}} \geq R_{\text{target}} \\
& SNR(T_{\text{pulse}}, d) \geq SNR_{\text{target}}(N, T_{\text{pulse}}, C_r)
\end{aligned} \tag{2.20}$$

The optimization results for the maximum data rate and energy efficiency are shown in Fig. 2.13 and Fig. 2.14 respectively. The results correspond to the NLOS scenario. Again, the solid black line on Fig. 2.13 and Fig. 2.14 (on the left) corresponds to a static scheme that uses $N = 1$, $C_r = 1$, and $T_{\text{pulse}} = 1\mu s$. Unlike this static scheme, the proposed system can provide graceful tradeoffs in the data rate (or energy efficiency) as a function of link distance when the optimal mode is selected by the proposed gateway guided link adaptation strategy. Up to two orders of magnitudes optimal data rate variation is observed when the distance changes from 0.1 to $31m$. For a target data rate of 10 $k$bit/s, the proposed millimeter-scale system achieves the optimal energy efficiency in the range of $0.01 - 5.5nJ$/bit depending on the operating link distance.

From the maximum energy efficiency link adaptation result, Fig. 2.14, one can compute

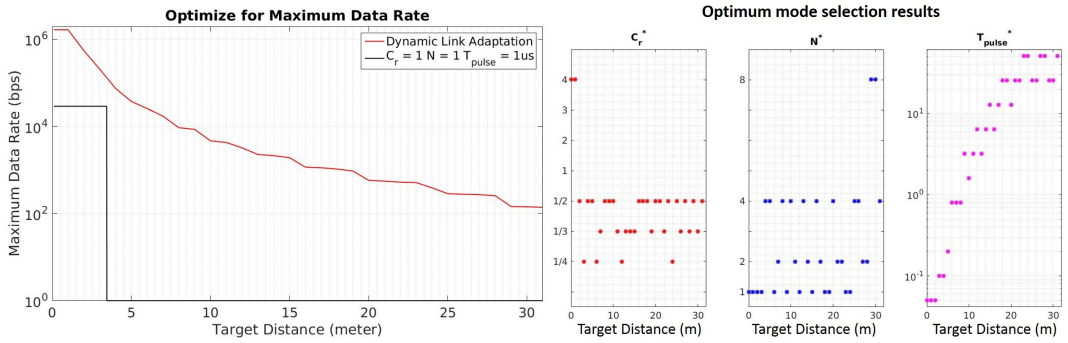


Figure 2.13: Maximum data rate objective link adaptation result

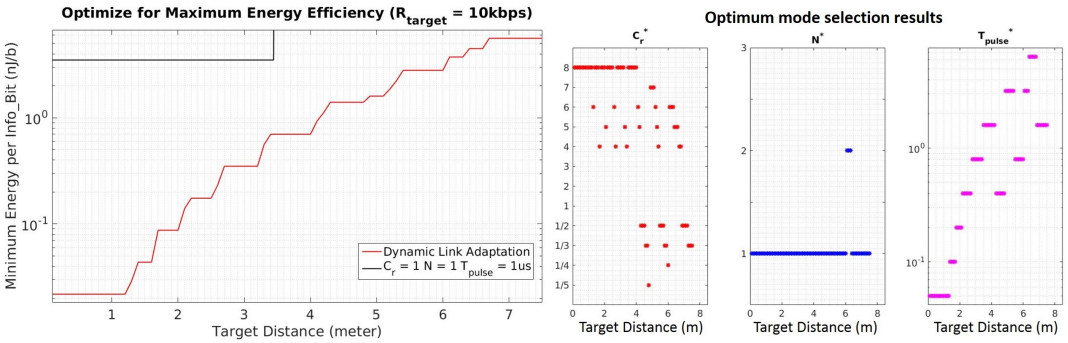


Figure 2.14: Maximum energy efficiency objective link adaptation result

the 'sustainable data rate'. To make the data rate sustainable, the energy consumption for communication has to be below the harvested energy level in average. The state-of-the-art solar energy harvester [19] reports $P_{\text{harvest}} \approx 1\mu W$ harvested power per mm^2 solar panel area. The 'sustainable data rate' is computed by P_{harvest}/E_b^* bit/s where E_b^* is the optimal energy efficiency per information bit obtained by solving (2.20). Any instantaneous data rate higher than the 'sustainable data rate' would deplete the energy from the battery and thus require duty-cycled operation. Results in Fig. 2.14 and $P_{\text{harvest}} = 5\mu W$ ($5mm^2$ solar pannel area) indicate the sustainable data rate is about 7.1 $k$bit/s for a $4m$ distance link in a NLOS scenario.

Fig. 2.13 and Fig. 2.14 (on the right) plot the best mode selection ($C_r{}^*$, N^*, and T_{pulse}^*) results for the maximum data rate and energy efficiency objectives respectively in the NLOS scenario. Notice non-monotonic behavior in the optimal C_r selection. The maximum data rate objective results show that $N > 1$ repetition of shorter pulses are advantageous to increase the data rate. On the other hand, $N > 1$ is rarely selected for the maximum energy efficiency objective. It indicates, for the same total energy per symbol, continuous energy draw (i.e., a single long pulse) is more energy efficiency than short pulse repetition for non-coherent communication.

2.4.3 Impact of System Parameters

In this section, we look into system parameters that need to be optimized at the system design time. Four system parameters are analyzed; the antenna size (r_{ant}), carrier frequency (f_c), battery current (I_{bat}), and reservoir capacitor size (C_{res}). Since these parameters are not adjustable for dynamic link adaptation, a special attention has to be paid to specify these parameters considering their impact on overall system performance. For this study, modulation-coding parameters (C_r, N, and T_{pulse}) are dynamically adapted for the optimal performance. All other parameters are set to default values specified in Table 2.3, unless specified otherwise.

The antenna size is the most critical system parameter that dominates the overall system volume (see Fig. 2.2). Smaller antenna sizes are certainly attractive to keep the system volume minimized. However, the radiation efficiency rapidly drops (see Fig. 2.3) as the millimeter-scale antenna size decreases. The left plot in Fig. 2.15 shows that $r_{ant} = 1, 1.5,$ and $2mm$ antennas can provide the maximum distance of 21.4, 31.3 and $41m$ respectively for the target data rate of 100 bits/s in a NLOS scenario with link adaptation when all other parameters are fixed as in Table 2.3. The right plot in Fig. 2.15 also confirms that increasing the antenna size is an obvious way to significantly improve the energy efficiency.

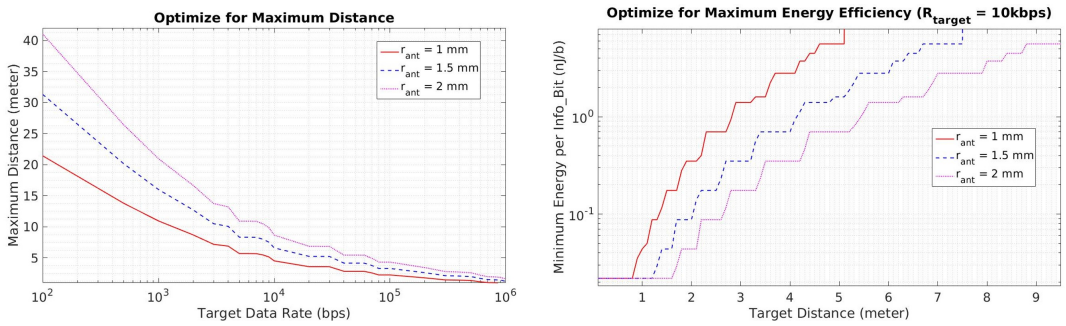


Figure 2.15: Impact of antenna size on maximum distance (left) and maximum energy efficiency (right) objectives

The carrier frequency (f_c) affects the system performance via the antenna efficiency as well as wall penetration/pathloss characteristic. The impact of the carrier frequency in LOS and NLOS settings highly depends on how the wall penetration/pathloss is modeled. Fig. 2.16 shows the maximum distance link adaptation results obtained from different carrier frequency settings in both LOS and NLOS settings using our model (2.1). It implies that $\geq 5GHz$ operation is preferred if the application mostly targets LOS scenarios. For NLOS indoor operation that follows our propagation loss model, $1GHz$ operation outperforms higher carrier frequency options, although the millimeter-scale antenna efficiency at $1GHz$ is very poor ($< 1\%$, see Fig. 2.3).

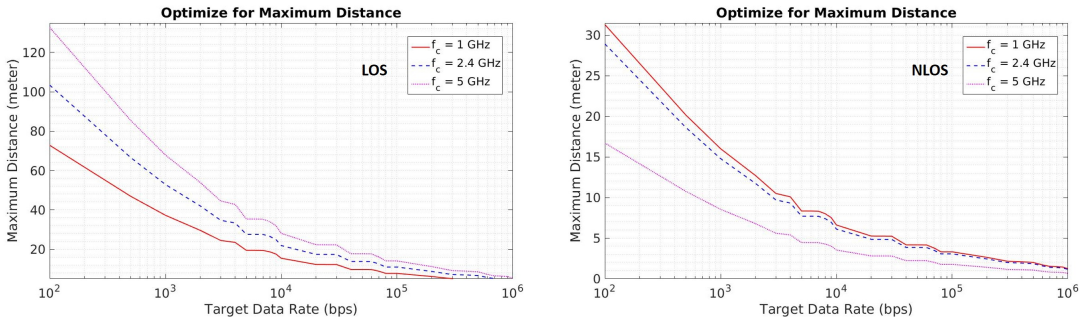


Figure 2.16: Impact of carrier frequency on maximum distance objective (left:LOS, right:NLOS)

In the proposed system, the thin film battery continually charges the reservoir capacitor. The battery current determines the recharging time T_{charge} via (2.15). A larger battery typically has a lower internal resistance, thus allows a higher battery current. Since T_{charge} is reduced with a higher battery current, higher energy per pulse can be drawn from the capacitor to increase the link distance while maintaining the same data rate. For a given link distance target, the link adaptation strategy can utilize the additional battery current to reduce the energy per data bit. The impact of various battery current levels is shown in Fig. 2.17.

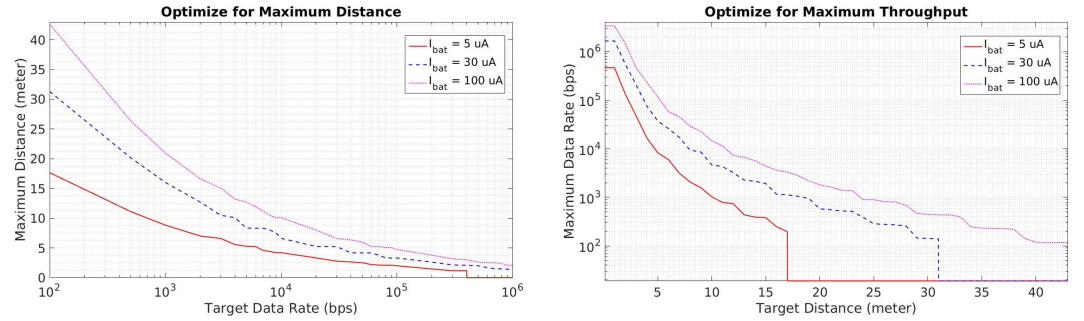


Figure 2.17: Impact of battery current on maximum distance (left) and maximum data rate (right) objectives

Finally, we inspect the impact of the reservoir capacitor, the energy buffer to power transceiver circuits during active short pulse transmission. Fig. 2.18 shows the impact of increasing the capacitor size C_{res}. For a short distance, the link adaptation system selects shorter pulses to maximize data rate, satisfying the SNR requirement only using partial energy stored in C_{res}. In this case, the capacitor size is not limiting the system performance. The larger capacitor starts to make difference for long distance operations ($> 16m$) when target data rate is low (e.g., 100s bit/s). For this case, a larger C_{res} enables higher energy per pulse, thus realizes a higher SNR. Given a constant I_{bat}, depleting more energy from

a larger capacitor would result in a longer charging time T_{charge}, potentially lowering the data rate if the link adaptation was disabled. In fact, the optimal link adaptation improves the overall data rate as shown in the right plot on Fig. 2.18 by applying higher coding rate utilizing the increased energy per pulse from a larger capacitor.

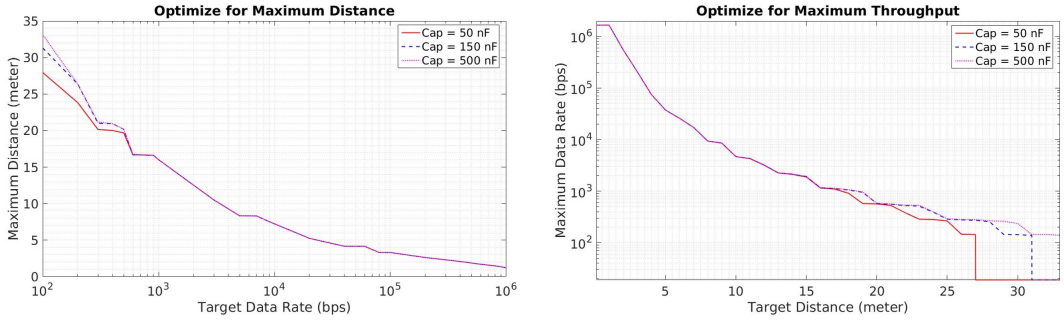


Figure 2.18: Impact of reservoir capacitor on maximum distance (left) and maximum data rate (right) objectives

In summary, for the maximum utilization of the millimeter-scale system dimension, the system designer must quantify the impact of each design parameter and purposefully determine the size of the antenna, battery, and energy reservoir capacitor along with the trasceiver/sensor/processor integrated circuits to be integrated in a ultra-small form-factor (Fig. 2.2). When the overall system dimension is fixed, increasing the size of one component would inevitably limit the size of the other. The link adaptation optimization framework and its results shown in this section provide a guideline for the designer to foresee the impact of critical design parameters in realizing a highly energy-optimized wireless communication system with a millimeter-scale form-factor constraint.

2.5 Demo System

The semi-realtime demo system is set as shown in Fig. 2.19. The gateway is demoed by a laptop connecting to an USRP. The millimeter sensor node transmits the packet with a pre-known synchronization header following by unkown data. The gateway is always on to listen to the packet and decode the unknown data.

The upper signal in Fig. 2.20 is the raw signal captured in the USRP frontend, and it is deeply hidden in the noise. The lower signal illustrates the synchronization result. The gateway identifies the synchronization header successfully and they are highlighted by green circles.

Fig. 2.21 illustrates the synchronization process in a detailed way. Similar to Fig. 2.20, the blue stem is the raw signal received and the red stem shows the matched filter outputs

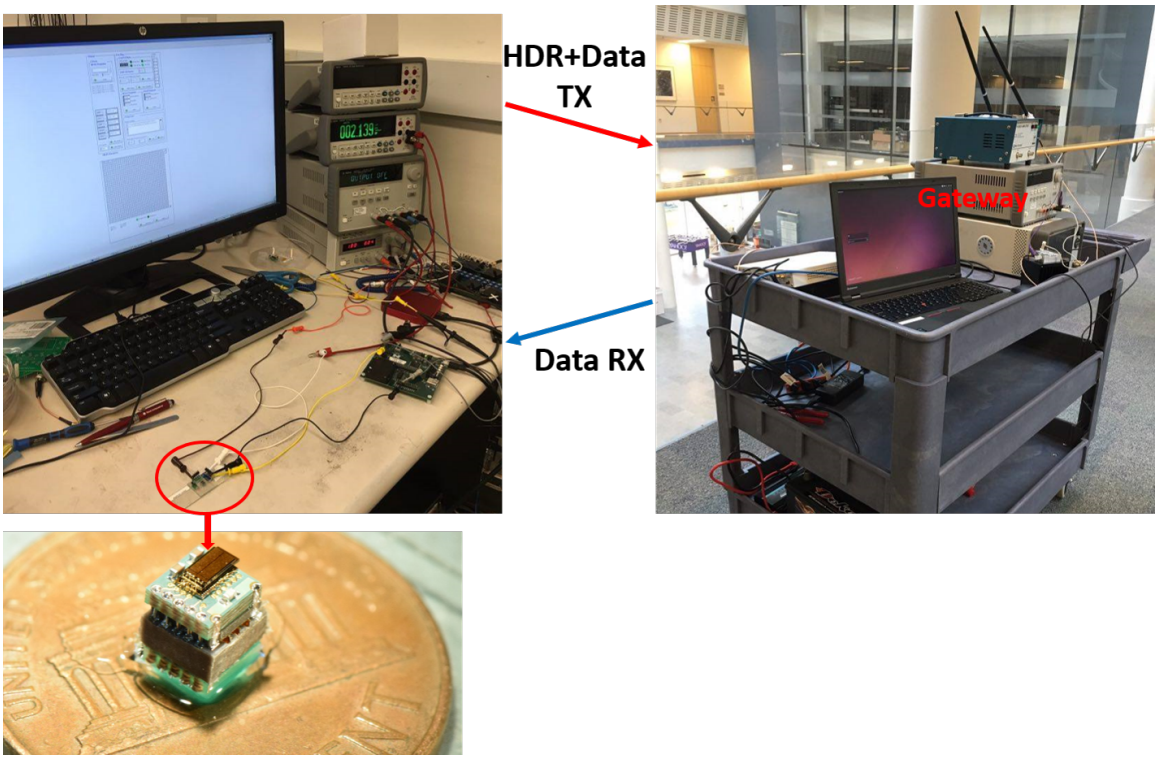


Figure 2.19: The demo system setting.

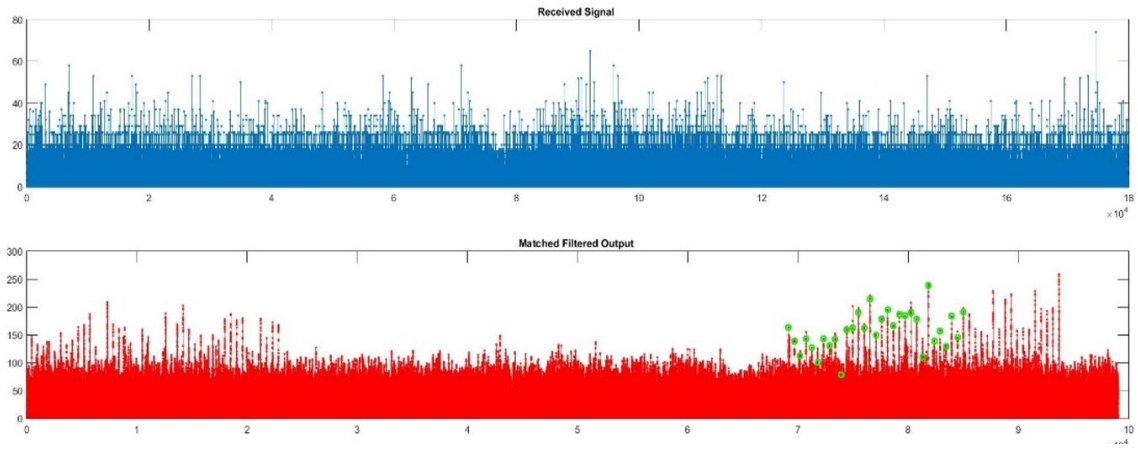


Figure 2.20: The packet is deeply hidden in the noisy raw received signal (upper). The packet is still recognized by the gateway and the packet header is highlighted by green circles (lower).

with synchronization header highlighted in green circles. The bottom left figure reveals the peak correlation value across time hypotheses (only part of the time hypotheses are shown) at the peak frequency hypothesis. The bottom right figure shows the peak correlation value occurs at 39th candidates across all frequency hypotheses at the peak time hypothesis.

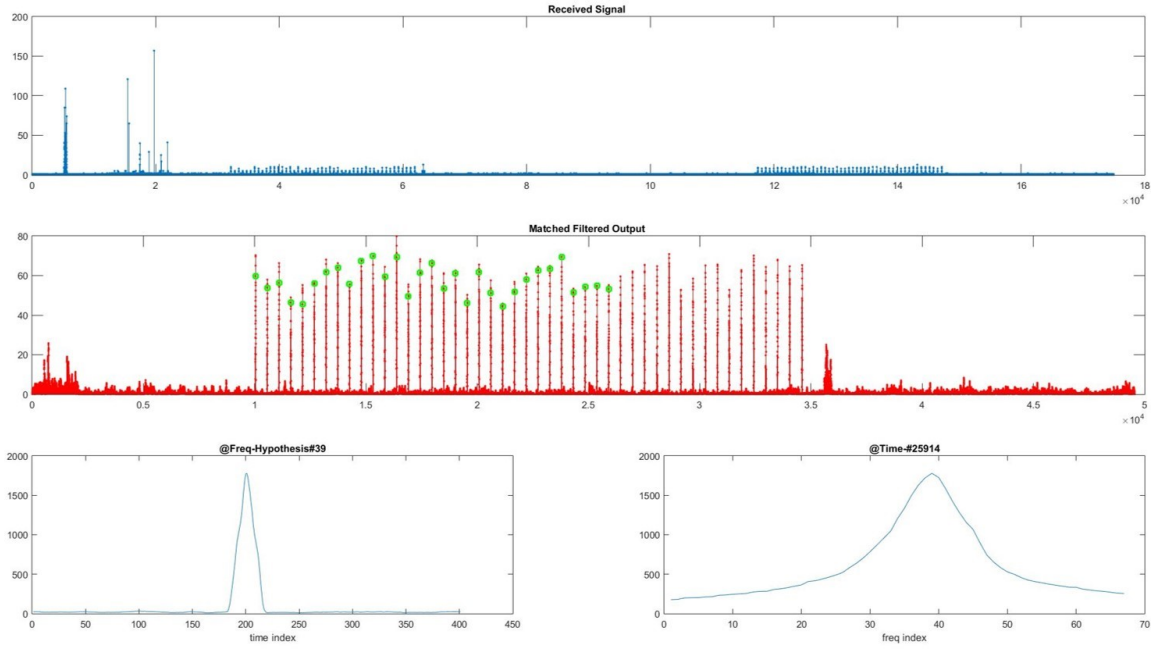


Figure 2.21: A detailed illustration of synchronization result.

2.6 Conclusion

In this work, an energy-autonomous, self-contained wireless communications system that optimally utilizes the scarce energy/power resource in an ultra-small millimeter-scale sensor node is presented. A cross-layer system-level optimization framework is proposed to jointly optimize various system parameters including modulation, coding scheme, synchronization protocol, RF/analog/digital circuit specifications, data rate, carrier frequency, antenna efficiency, etc. Based on the comprehensive system model, the dynamic link adaptation problems are formulated to maximize transmission distance, throughput, and energy efficiency for various operating scenarios. The simulation results of the link adaptation protocol show the significant benefit of dynamically adapting to optimum system parameters. The impact of pre-silicon system parameters including antenna dimension, carrier frequency, battery current, and reservoir capacitor size is presented to guide system designers toward the energy-optimized communication system for ultra-small IoT sensor nodes.

CHAPTER 3

A Low Power Software-Defined-Radio Baseband Processor for the Internet of Things

In this work, we define a configurable Software Defined Radio (SDR) baseband processor design for the Internet of Things (IoT). We analyzed the fundamental algorithms in communications systems on IoT devices to enable a microarchitecture design that supports many IoT standards and custom nonstandard communications. Based on this analysis, we propose a custom SIMD execution model coupled with a scalar unit. We introduce several architectural optimizations to this design: streaming registers, variable bit width datapath, dedicated ALUs for critical kernels, and an optimized flexible reduction network. We employ voltage scaling and clock gating to further reduce the power, while more than a 100% time margin has been reserved for reliable operation in the near-threshold region. Together our architectural enhancements lead to a $71\times$ power reduction compared to a classic general purpose SDR SIMD architecture.

Our IoT SDR datapath has sub-mW power consumption based on SPICE simulation, and is placed and routed to fit within an area of 0.074mm^2 in a 28nm process. We implemented several essential elementary signal processing kernels and combined them to demonstrate two end-to-end upper bound systems, 802.15.4-OQPSK and Bluetooth Low Energy. Our full SDR baseband system consists of a config- urable SIMD with a control plane MCU and memory. For comparison, the best commercial wireless transceiver consumes 23.8mW for the entire wireless system (digital/RF/ analog). We show that our digital system power is below 2mW, in other words only 8% of the total system power. The wireless system is dominated by RF/analog power comsumption, thus the price of flexibility that SDR affords is small. We believe this work is unique in demonstrating the value of baseband SDR in the low power IoT domain.

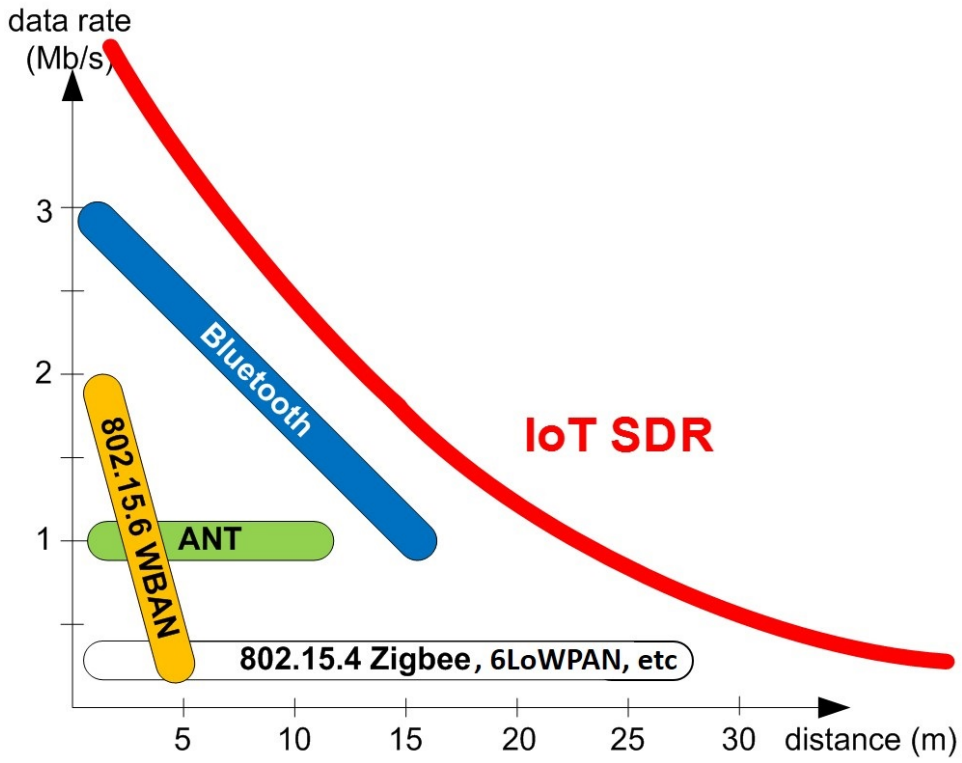


Figure 3.1: **The Application Domain of the IoT SDR.** IoT standards support either long distance or high data rate to limit power consumption. An **IoT SDR** can be software defined to operate at points in the region under the red line.

3.1 Introduction

The Internet of Things (IoT) [36] is an emerging concept which envisions that physical objects or "things" comprised of digital hardware, software, and sensors are connected to the Internet. It allows, among other things, remote data collection from sensors, remote management, and automatic data and control between devices. The IoT concept has been placed third among top ten strategic technology in the past year [37]. And the number of "things" that will be connecting to the Internet will exceed PCs and smartphones [38].

Central to this vision is wireless connectivity because it may often be the case that connecting them to LANs is infeasible. Furthermore, many of these devices will depend on batteries and/or local sources of energy such as solar power. As such the wireless connectivity should not add unnecessarily to the overall power consumption.

Several wireless standards have been proposed for the IoT communications. Figure 3.1 identifies four of the most popular. A common issue is the trade-off between distance and data rate because of the need to limit power consumption. Different standards are designed for different operating scenarios, as Figure 3.1 shows.

3.1.1 SDR is Promising for IoT Applications

Software-Defined Radio (SDR), in which the communications algorithms are executed on a programmable platform rather than a traditional ASIC solution, is promising for IoT devices for several reasons.

Multi-Standard Scenario. The market for the IoT is full of different standards; some very popular standards are shown in Figure 3.1. IEEE 802.15.4 [2] is a standard for low-cost, Low Rate Wireless Personal Area Networks (LR-WPAN), which is the basis of many existing IoT upper layer protocols such as 6LoWPAN, ZigBee, ISA100 and so on. It is widely used in wireless sensor network applications, providing comparatively long distance transmission and robust communications.

Bluetooth [4] is another popular standards series for personal area networks, and is targeted for high data rates over short range. Bluetooth is widely adopted in mobile devices. ANT [39] is a short range wireless protocol, providing low data rates (constant 1Mbps). The ANT radio transceiver is a proprietary specification used mainly for sports equipment. Finally, IEEE 802.15.6 [3] Wireless Body Area Network (WBAN) is a standard that supports communications for low power devices such as healthcare equipment that are near or inside the human body.

Besides the most popular standards listed in Figure 3.1, there are many other less popular wireless technologies that are suitable for other IoT scenarios, making it more and more challenging to choose one wireless connectivity technology over another for a given IoT application [40]. Ideally, the IoT devices should have the ability to support multiple standards— SDR enables this. In fact, if multiple standards require multiple ASICs, the solution can be extremely area inefficient. Finally, the processor can be programmed to operate as both a receiver and a transmitter.

SDR supports on-the-fly updates to standards. Any change can be achieved via software/firmware download to meet the particular demand of the specific IoT application. This feature becomes more interesting in the IoT domain because standards change within a relative small performance region in terms of data rate and distance. In contrast to the standards evolution in wide-band communications, such as LTE, there is little data rate or distance difference for IoT standards from generation to generation. An SDR architecture, which has the scalability to achieve the performance upper bound, will be able to capture standard updates on the same hardware platform while saving a lot effort compared to ASIC design and fabrication.

SDR enables graceful data rate/distance trade-offs. In addition to supporting standards updates, by allowing nonstandard protocols on the baseband processing, an SDR architecture will bring more degrees of freedom in the space of data rate, distance and en-

ergy efficiency. By dynamically changing configurations, signal bandwidth and modulation parameters on the SDR platform, the IoT devices can adjust data rate to extend operating distance with the same power budget or achieve higher energy efficiency when operating scenario requires shorter distance and/or lower data rate that are defined in a particular wireless standard.This also allows adaptation to changes in channel conditions.

The overhead of SDR can be kept small in the IoT domain. Typically, the power consumption of the entire digital module is small compared to the RF/analog power in IoT devices. Meanwhile, power reduction for RF and analog is much harder because a power-demaning amplifier operating at RF carrier frequencies (typically in GHz) is necessary to overcome the over-the-air signal power loss that follows an inverse square law, and RF frequency local oscillator consumes significant power regardless of the modulation scheme. The fact that the RF/analog dominates actually exposes a great opportunity for SDR as long as an SDR solution still remains a fairly small percentage compared to the RF/analog. As we will show, the cost of our proposed solution stays within a minor portion (8%) of the overall power budget when including RF and analog portions.

3.1.2 Feasibility of SDR in the IoT Domain

Several observations from IoT communications expose the possibility to support SDR with little overhead.

Similar Datapaths for Different Standards. Key functions are very similar in multiple standards. For example, Bluetooth [4], ANT [39] and 802.15.4g FSK-SUN [41] all employ FSK modulation. 802.15.6WBAN [3] and 802.15.4-OQPSK [2] are both based on quadrature linear modulations. A very similar datapath can be applied in many different standards.

Key Kernels Share Computation Patterns. There are some computationally intensive baseband processing kernels in a radio transceiver, such as Synchronization and Finite Impulse Response (FIR) filters, that every communications system needs to include. These kernels have very similar computation patterns, which parallelize in a straightforward manner. These common kernels allow the development of a general and computationally efficient SDR processor for low power IoT applications.

Key Kernels Dominate the Power. Since there are key kernels that dominate the power in the communications datapath, the tight power budgets can be met by focusing on these kernels on an SDR platform.

3.1.3 Design Challenges and Contributions

The challenges of designing a configurable IoT SDR processor that supports multiple standard and nonstandard communications are the strict power and area constraints. The IoT devices are, and will be, in small, power limited everyday objects. To keep enough flexibility while maintaining the power/area of SDR in a small region compared to the whole system is the key challenge.

In this chapter, we present the IoT SDR architecture, consisting of a custom Single Instruction Multiple Data (SIMD) Unit and a Scalar Unit. Several low power techniques have been employed so as to meet the tight power budget and area constraints. The contributions of this work are:

1. Identifying common computation patterns in IoT wireless standards and analyzing them for parallelism and bit precision.

2. Designing a low power architecture that achieves ASIC comparable SIMD efficiency for dominant kernels, in which its bit width is configurable (to as small as 4 bits) throughout the SIMD datapath, complex/real dedicated ALUs, and reduction networks. It also supports configurable streaming registers.

3. Optimizing the design by exploring the system trade-offs in area and power with accurate post-layout analysis that considers low power techniques such as voltage scaling and clock gating.

4. Fully evaluating the architecture by examing elementary signal processing kernels, and combining these kernels to build two representative upper-bound end-to-end standards, specifically 802.15.4-OQPSK and Bluetooth Low Energy, suitable for exploring the design of an SDR processor.

We demonstrate baseband SDR is possible even in a low power IoT domain.

3.2 Baseband Processing of IoT Communications

The baseband processing at the receiver is more complicated than at the transmitter [42] for two reasons: the uncertainty of the packet arrival time and symbol boundary; and the noise introduced by the channel. Accounting for this requires extra computation steps to be performed. For this reason, we limit a discussion to the receiving functions, because in a configurable setting the transmitter just requires us to execute a subset of the receiver communications kernels.

At the receiver, the FIR filters and the Synchronization are the most computationally intensive kernels. And Figure 3.2 illustrates a general communications system.

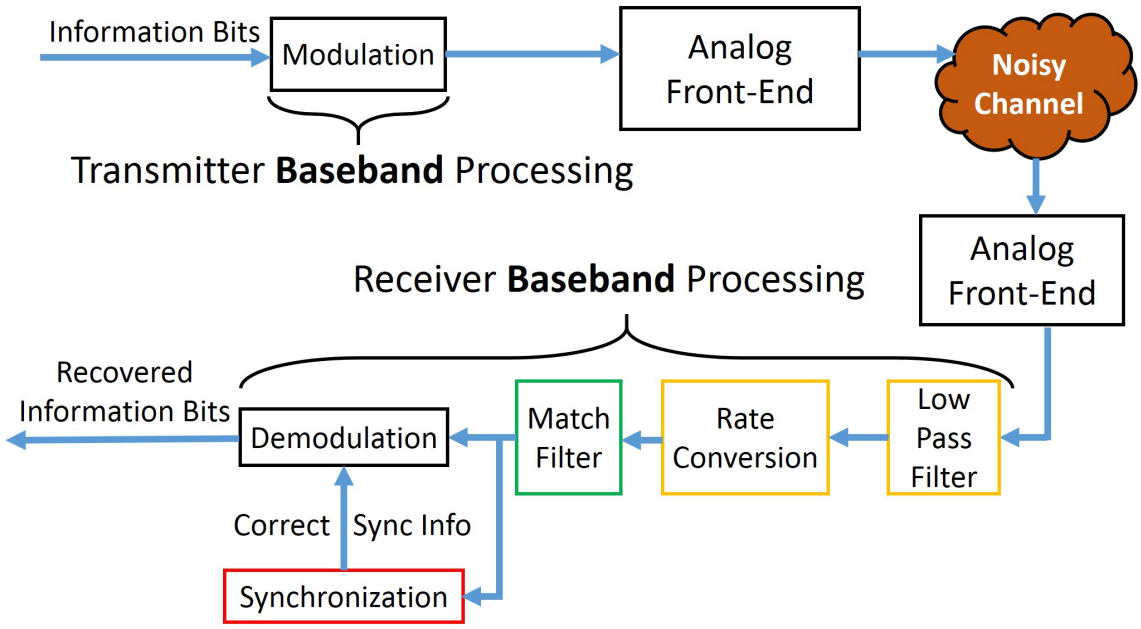


Figure 3.2: **Baseband Processing of an IoT Communications System.** The receiver is more complicated than the transmitter. It adds a Low Pass Filter, Matched Filter and Synchronization which are computationally intensive.

3.2.1 FIR Filter Computation Characteristics

There are two basic FIR filters in IoT communications systems shown in Figure 3.2: a Low Pass Filter (LPF) and Matched Filter. The Low Pass Filter (or channel selection filter) needs to be applied to remove out-of-band noise and adjacent channel interferences, and to avoid aliasing. Rate conversion, discussed in Section 2.4, is used to down convert the signal from the ADC output rate to the target over-sampling rate, discussed in Section 2.3. The Match Filter is used to maximize the Signal-to-Noise Ratio (SNR). The computation of the FIR is a convolution as shown in eq. (1). The length of the filter is indicated by L and $c[j]$ is the j-th coefficient of the FIR filter. The value of $c[j]$ is typically a real number whereas r is typical a complex number.

$$y[i] = \sum_{j=0}^{L-1} r[i+j]c[j] \tag{3.1}$$

An illustration of FIR filters are shown in Figure 3.3. Three observations can be made from eq. 3.1. First, the computation is streaming in nature. Second, the multiplications can be vectorized. Finally, a vector reduction unit is needed to compute the summation efficiently.

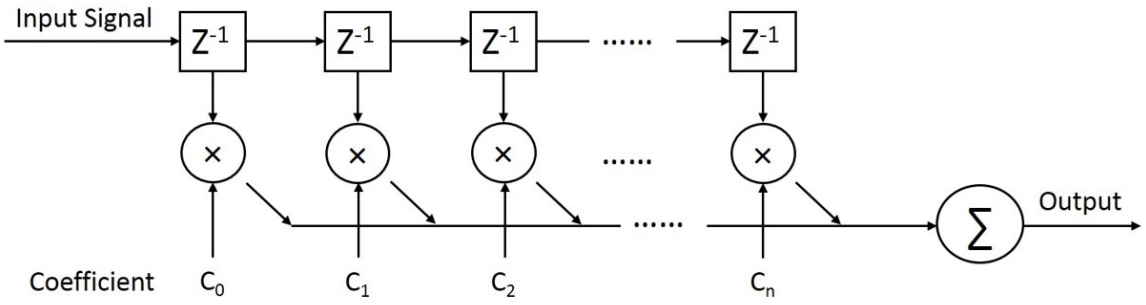


Figure 3.3: **FIR filter Characteristics.**

3.2.2 Synchronization Computation Characteristics

The major task of Synchronization is to find the beginning of the packet in the received signal. The Data-Aided method [43] is widely used in this process for many communications standards. A known signal is used as a reference or pilot signal in the header of each transmitted packet. The receiver correlates the known signal pattern with the incoming data, and measures the difference as shown in eq. 3.2.

$$\tau = argmin_t \left\| \underline{r}[t] - \underline{p} \right\| \tag{3.2}$$

$$\text{where } \underline{r}[t] = [r[t], r[t+1], ..., r[t+P-1]] \; \underline{p} = [p_0, p_1, p_2, ..., p_{P-1}]$$

The received vector at time t is $\underline{r}[t]$, and $\underline{p}$ is the reference signal vector. P is the vector length. Eq. 3.2 can also be reduced to eq. 3.3, where p_j^* is the conjugate of p_j.

$$\tau = argmax_t \left| \sum_{j=0}^{P-1} r[t+j] p_j^* \right| \tag{3.3}$$

Eq. 3.3 is the well-known Maximum Likelihood (ML) timing estimation [44]. Eq. 3.2 can be written as eq. 3.4 when both $\underline{r}[t]$ and $\underline{p}$ are real-number vectors.

$$\tau = \underset{t}{argmin} |\underline{r}[t] - p| = argmin_t \sum_{j=0}^{P-1} |r[t+j] - p_j| \tag{3.4}$$

The "max" (or "min") operation does not need to be triggered until the correlation value is above an empirical threshold to save unnecessary computation. The correlation in eq. 3.3 and eq. 3.4 is the computationally intensive operation since τ needs to be reevaluated for every incoming sample.

An illustration of synchronization is shown in Figure 3.4.Similarly to FIR filters, both eq. 3.3 and eq. 3.4 are streaming in nature, vectorizable, and require a reduction operation.

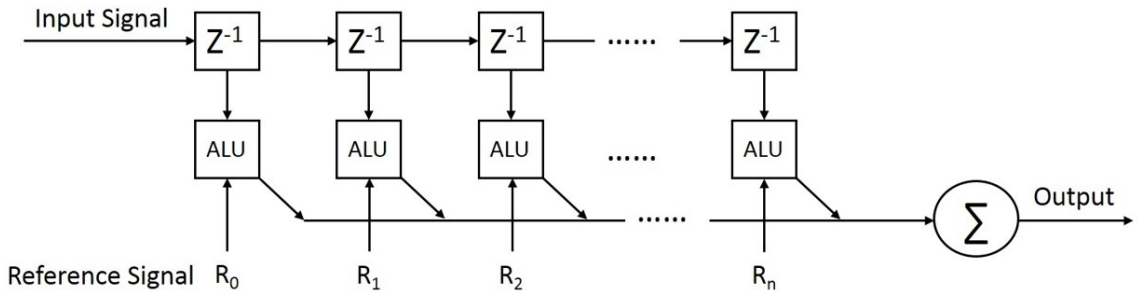


Figure 3.4: **Synchronization Characteristics.**

In addition, max and min operations are required which can be handled in a non-vector unit.

3.2.3 Over-Sampling Rate

The FIR and Synchronization kernels are computationally intensive because they over-sample. The over-sampling is required to accurately estimate timing and provide maximum SNR. The over-sampling rate is usually $4\times$ higher than the data symbol rate. Take as an example Classic Bluetooth running at 1Mbps. The signal is modulated and up-sampled to 4MSPS (Mega Samples Per Second) before it is converted to an analog waveform. At the receiver, the FIR filters are always active at 4MSPS as long as the IoT node is in receiving mode. The Synchronization is also run at the over-sampling rate of 4MSPS until the start of the packet is correctly found. After the synchronization is acquired, the demodulation process can be run at the sample rate of 1MSPS. This explains the reason that FIR and Synchronization are the computationally intensive operations. This also indicates that the main architectural datapath be able of producing results at the over-sampling rate.

3.2.4 Rate Conversion

The ADC output rate is typically higher than the over-sampling rate because using low Intermediate Frequency (IF) architectures to avoid the analog circuit flicker noise around DC and high-order channel selection filter in the analog domain is area/power inefficient. Therefore, rate conversion is introduced to down-convert from the high ADC output rate to the over-sampling rate. The other purpose of rate conversion is to support various signal bandwidth options on-the-fly. The lower the bandwidth, the smaller the integrated noise power. Hence, we can achieve longer communications distance with lower data rates (smaller bandwidth). The LPF and rate conversion in Figure 3.2 can be merged into one step by employing Streaming Registers, discussed in Section 3.2.1. In this way, the com-

putation for those outputs that will be discarded by rate conversion can be avoided. Thus, the LPF and Match Filter are run at the over-sampling rate.

3.2.5 Criticality of Synchronization

The transmission pattern of the IoT devices magnifies the overhead of the synchronization. Small IoT devices often wake up periodically to transmit or receive a small amount of data and return to sleep mode in order to save power [45]. Every time the node needs to receive data, synchronization is required. Even when the node wakes up to transmit data, it has to wait to receive an ACK, which keeps the Synchronization kernel busy until the ACK occurs. Moreover, the computation effect of the synchronization cannot be amortized since the length of a data packet, which is defined in the standards, is fairly small.

3.2.6 Communications Datapath Bit Width

We conduct system simulations with different bit width datapaths. The influence of bit width is shown in Figure 3.5. The communication performance measurement on the vertical axis is Bit Error Rate (BER). BER measures the difference between the information bit sequence at the transmitter and the recovered bit sequence at the receiver; for this experiment on the 802.15.4-OQPSK system, at least 10^4 error bits are collected for each point in Figure 3.5. The horizontal axis is SNR, which indicates the relative ratio between useful signal power and noise power. Given the maximum packet length of 802.15.4 is 133 Bytes [2], a target BER of 10^{-4} will result in 0.1 packet error rate. Therefore we choose 10^{-4} as a comparison point.

The middle and bottom curves in Figure 3.5 indicate that 8-bit communications datapath (middle) can capture almost the same precision as the floating point datapath (bottom). The 2-bit Maximum Likelihood (ML) synchronization estimator (top), as described by eq. 3.3, in which only the sign information of complex value is kept, is also implemented and compared with the full 8-bit ML synchronization estimator. The difference is less than 1dB, indicating that it is sufficient to achieve target communication performance.

3.2.7 Signal Processing Characteristics and Design Implication

To conclude the above discussion, the baseband processing of IoT communications has the the following characteristics: (a) Dominant kernels are all vector operations with different vector sizes followed by a reduction network; (b) Data comes in streaming fashion; (c) The

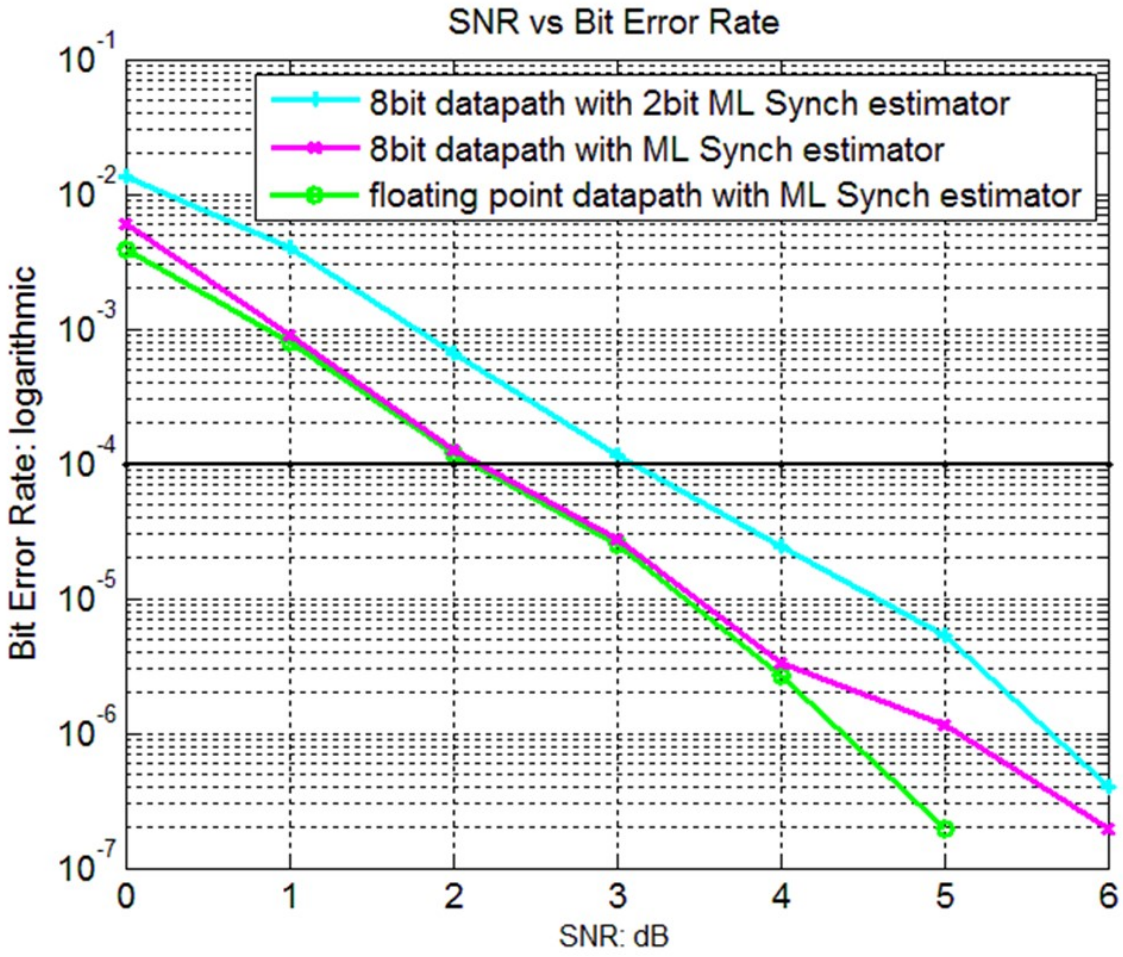


Figure 3.5: **Signal-to-Noise Ratio (SNR) vs Bit Error Rate (BER) for Different Bit Width Implementation.** The 8-bit communications datapath captures almost the same precision as the floating point datapath does. The 2-bit Synchronization estimator is sufficient to achieve the target BER.

datapath can be reduced to very small bit widths without sacrificing appreciable performance; (d) Although kernels are running at the over-sampling rate, the required frequency is still low (a few MHz) due to the low target data rate of IoT communications.

In the next section, we provide the design of an IoT SDR baseband processor, detail the architectural supports and optimizations to achieve flexibility and to achieve close to ASIC efficiency for critical kernels. We will show in Section 4.3 that the computationally intensive kernels can be efficiently mapped onto this architecture while the non-critical kernels can also be mapped easily.

3.3 The IoT SDR Architecture

In this section, we will present the design of our IoT SDR architecture. The architecture relies on novel structures—streaming registers, an optimized flexible bitwidth vector unit that supports both real and complex numbers, a multi-output reduction network, and a scalar unit—along with several effective techniques such as voltage scaling and clock gating to achieve a low power configurable solution.

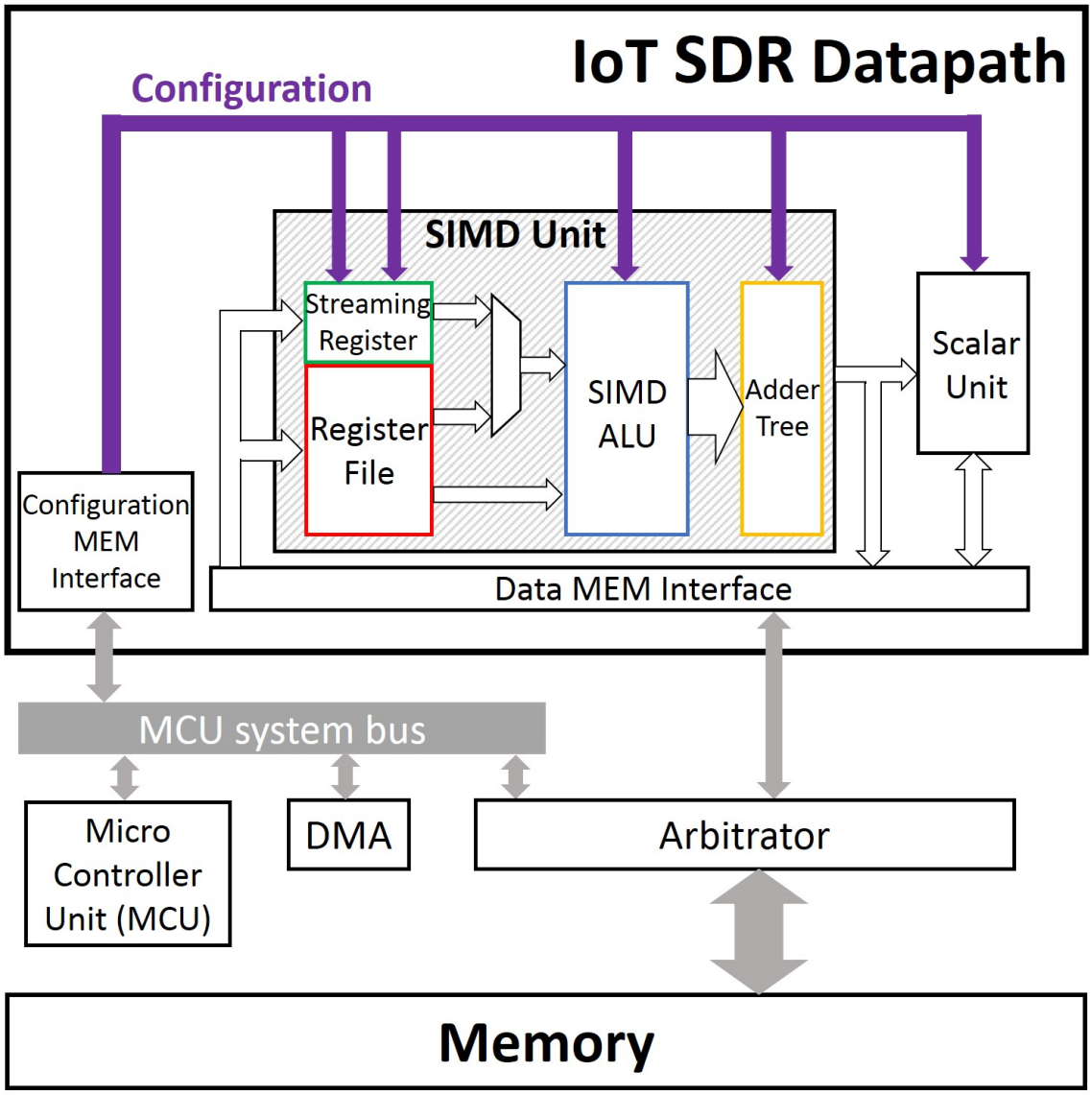


Figure 3.6: **IoT SDR System Architecture.** The IoT SDR is controlled by an MCU; data is shared by the MCU and IoT SDR via memory.

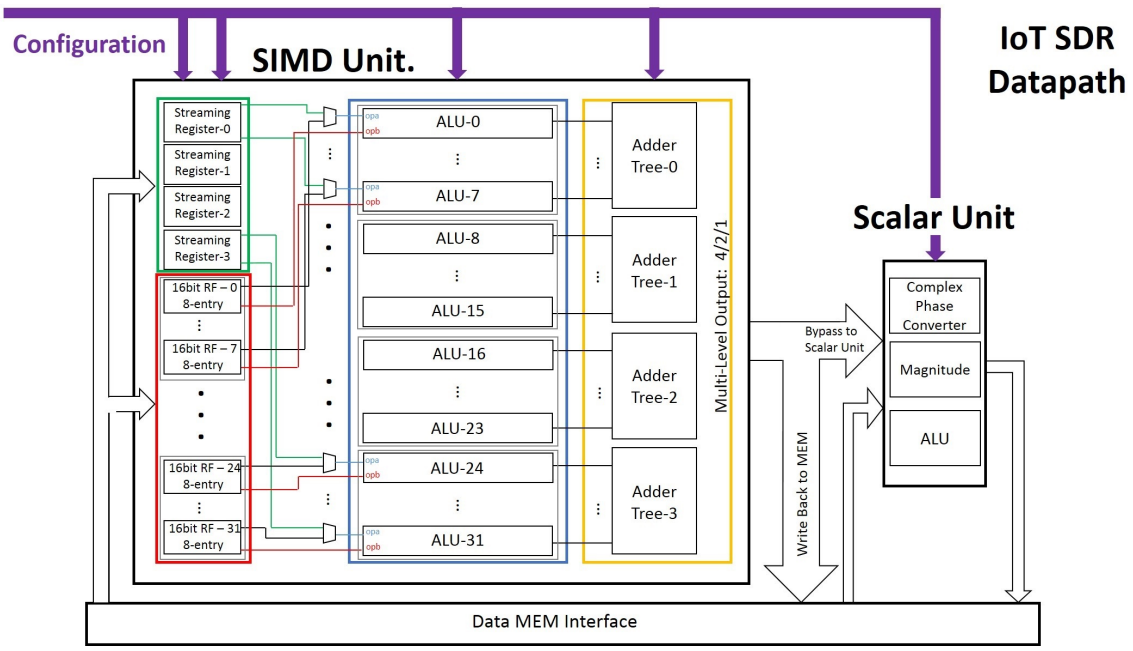


Figure 3.7: **The Microarchitecture of the IoT SDR Datapath.** An IoT SDR datapath comprises one **Scalar Unit** and one **SIMD Unit** consisting of Streaming Registers, a Register File, SIMD-ALUs and an Adder Tree.

3.3.1 System Architecture

The system architecture is presented in Figure 3.6. The IoT SDR datapath is directed by a programmable Micro Controller Unit (MCU). The MCU controls the Direct Memory Access (DMA) to copy the configurations for the SDR datapath from memory via the MCU system bus. Data communication between the MCU and the SDR datapath is through memory. Within the SDR datapath, data communication is handled by a bypass network to avoid unnecessary memory accesses.

Other than configuring the SDR datapath to do computationally intensive kernels, the MCU is also responsible for control-related computations such as the max and min operation in eq. 3.3 and eq. 3.4 of the synchronization stage.

3.3.2 Microarchitecture

The microarchitecture of this IoT SDR datapath is illustrated in Figure 3.7. There are two computational units: the SIMD Unit and the Scalar Unit.

3.3.2.1 SIMD Unit

The SIMD Unit is responsible for computationally intensive vector and reduction operations. There are a total of 32 lanes in this SIMD Unit, which can be grouped into four bundles of eight lanes for small vector size operations. In this case, the 32 lanes are performing the same function, but can provide four outputs simultaneously. Each lane of the SIMD Unit takes two 16-bit operands. A flexible bit-width is supported in this design. Depending on the operand bit width, the vector size supported in one SIMD Unit is set accordingly. For example, when the operand is 8 bits, the maximum vector size is 64. When the operand is 16 bits, the maximum vector size is 32. The configurable SIMD lanes and variable bitwidth is illustrated in Figure 3.8. The design of the four important components: Streaming Registers, Register File, SIMD-ALU and Adder-Tree will be detailed in the following paragraphs.

The **Streaming Registers** are deployed to match the streaming nature of the incoming data. The vector computation in the SIMD Unit will load only one new incoming data and reuse the data from previous cycles, as expressed in eq. 3.1, eq. 3.3 and eq. 3.4.

The Streaming Registers can be configured as one 512-bit, two 256-bit or four 128-bit streaming registers to handle vector computations of varying length. Specifically, seven different streaming bit widths (0/8/16/24/32/48/64 bits) are supported. The microarchitecture of Streaming Registers is presented in Figure 3.9.

As we mentioned in Section 2.4, the low pass filter and rate conversion can be computed with only one configuration. Consider the down-sampling and low pass filtering as a typical utilization example. Suppose the down sampling factor is 2 with 8bits/sample. By setting the streaming bit width to 16 bits, and configuring the SIMD-ALU and Adder Tree for a filtering functionality, the output of the SIMD Unit is exactly the down-sampled and low pass filtered value. The seven streaming-widths enable 0/1/2/3/4/6/8 down-sample factors for 8bits/sample while also enabling 0/1/2/4 factors for 16bits/sample. Down-sampling factors larger than eight are not supported because the number of filter taps is constrained by the total lane size. Larger down-sample factors with a small number of taps will result in a poor low pass frequency response for anti-aliasing or out-of-band noise filtering. However, we can still realize a larger down-sampling factor filter by cascading multiple FIRs (e.g. 16=4$\times$4). Moreover, fractional resampling ($1.0 <$ down-sampling < 2.0) can also be realized by bypassing the outputs from the SIMD Unit to the Scalar Unit for further processing.

When an IDLE configuration is fed into the SIMD Unit, a 0-bit streaming width is set, and the value for the streaming registers will be held, as shown in Figure 3.10. During an IDLE configuration, neither the streaming registers nor the regular register file has switch-

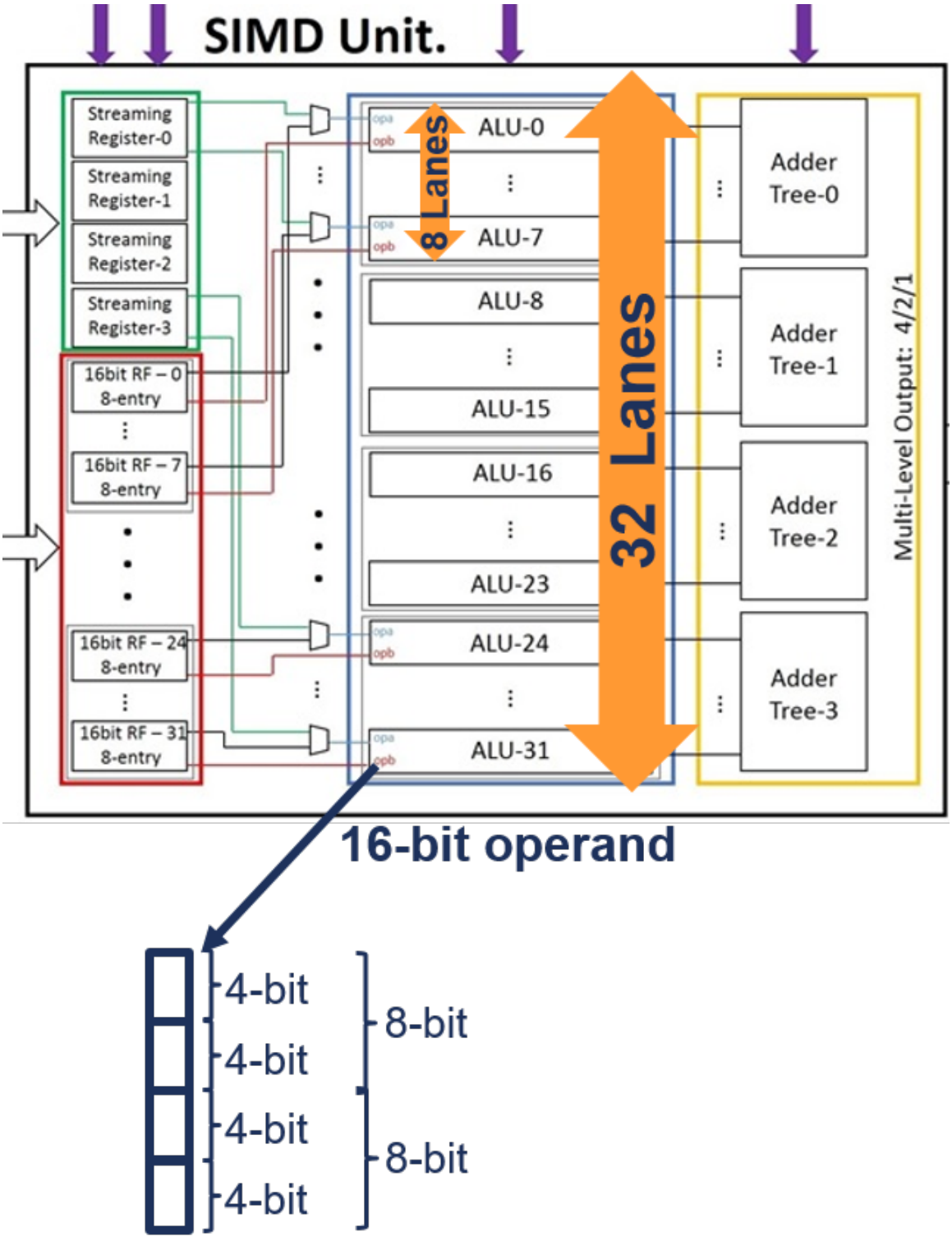


Figure 3.8: Both lanes and bitwidth are configurable in SIMD unit.

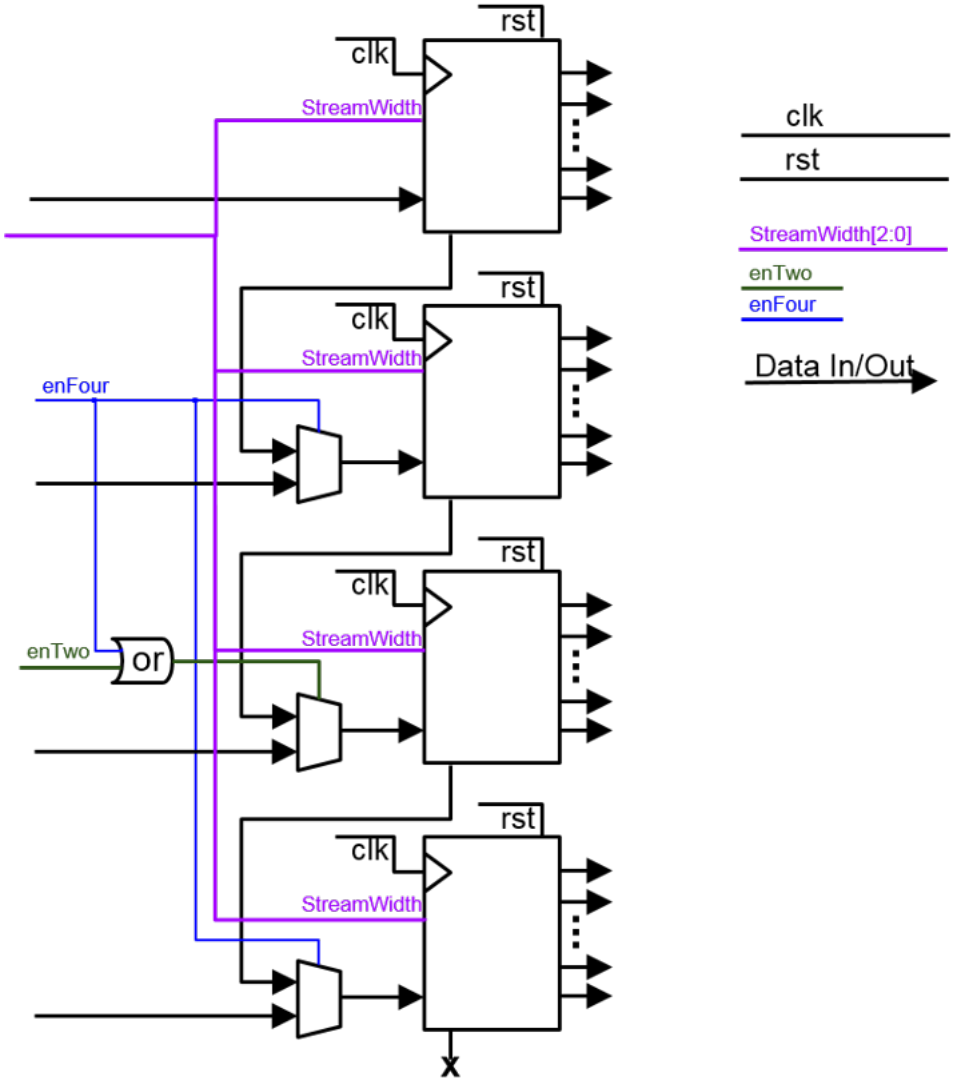


Figure 3.9: Streaming Registers Microarchitecture

ing activity, resulting in a nearly zero switching activity in the SIMD ALU and Adder Tree. The zero-width streaming IDLE minimizes the idle power significantly.

The streaming registers in this design are distinguished from earlier machines that have streaming data features [46][47] Our streaming registers are explicitly configured to provide data directly to ALUs with configurable bit precisions and vector sizes. With this novel structure, this architecture can support propriety but non-standard data rates, which enables graceful data rate and distance trade-off.

The IoT baseband datapath reduces the power by **10.4**× compared to the same datapath without Streaming Registers. This is due to the increased memory bandwidth and higher frequency required to meet the response time. This faster frequency ultimately limits the

Figure 3.10: A 0-streaming bit width is set by an IDLE configuration. During the IDLE configuration, the value for streaming registers will be held. Neither the streaming registers nor RF has switching activity, force all following combinational logic in a nearly zero switching activity.

voltage scaling level.

The **Register File (RF)** is implemented as an 8-entry 16-bit register for each lane. The RF is typically used to hold constants such as coefficients for filters. Eight entries mean at most eight different constant settings can be stored. To support the entire application, only four or five constants settings are needed. The necessary coefficients will be loaded into the RF at initialization. This strategy significantly reduces the re-configuring overhead when the SIMD Unit is time multiplexed by several kernels. Coefficients changing due to kernel switching can be easily realized by changing the source operand's register index.

The **SIMD-ALU** supports different bit width operations and several dedicated ALUs to accelerate computationally intensive kernels such as Synchronization. The SIMD-ALU will determine from the configurations whether to operate on two 8-bit real values, one 8-bit complex value, or two 4-bit complex values. As we mentioned in Section 2.6, reduced bit width computation is sufficient to achieve the target performance in terms of BER. The 4-bit operations are also handled within the 8-bit architectural datapath, improving throughput by $2\times$ with minimal area overhead.

Rather than regular ALU operations, dedicated merged configurations called "4-bit complex multiplication and add" and "subtract and add" as well as complex multiplication are specialized to accelerate the correlation operations in Synchronization. Without dedicated synchronization ALUs, each lane will save 44% of hardware, However, the system has to run at a higher frequency with larger memory bandwidth to support the same computation. The resulting system consumes **4.7**$\times$ more power than our proposed solution.

In order to reduce the computational pressure on the first level of the Adder Tree, we evaluate the truncation bits in the ALU. Each lane will take two 16-bit operands and provide

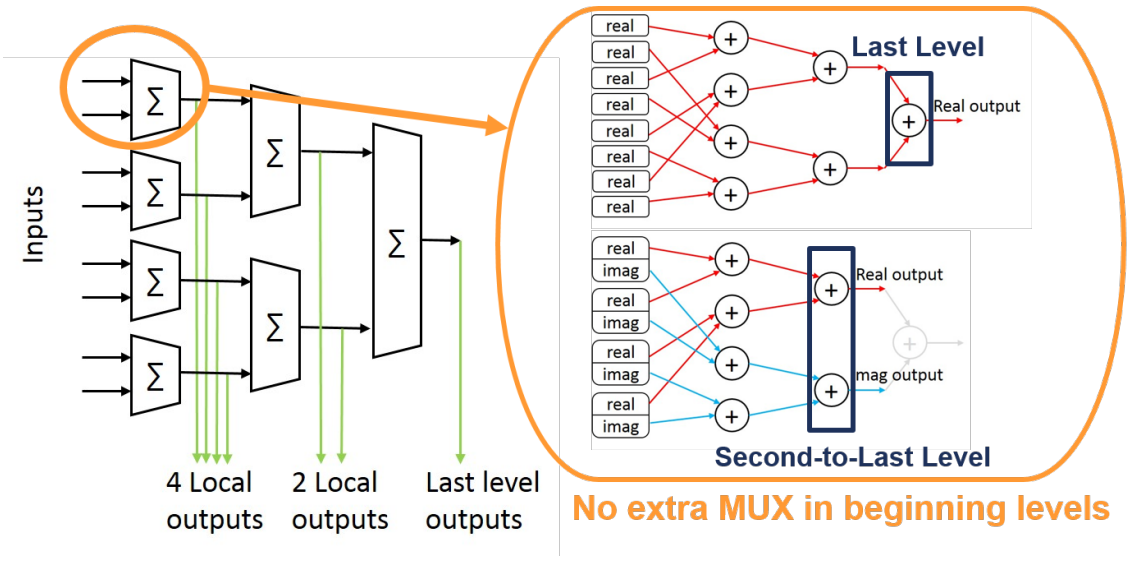


Figure 3.11: The optimized multi-level adder tree supports various vector size, parallel local outputs, and optimized complex summation. We interleaved the operands on the very first layer to avoid extra MUX in the beginning levels.

a total of 20 bits of output. This 20-bit output could be either two 10-bit or one 20-bit value.

The **Multi-Level Output Adder Tree**, as illustrated in Figure 3.11, is the reduction unit that is introduced to accelerate large summations. More features are used compared to previous adder-trees [48]. Our design is optimized to support flexible vector sizes, parallel local sum outputs and complex number summations. One, two, or four complex or real outputs can be supported by the Multi-Level Output Adder Tree.

In order to support the complex summation with the least number of multiplexers, we interleave the operands for the first layer adder, resulting in the top half of lanes adding the real part and the bottom half of lanes adding the imaginary part. Thus, if the input is complex, the result is selected from the second-to-last level outputs. If the input is real, the result is selected from the last level outputs.

Variable Bit-Width minimizes the power consumption and maximizes SIMD utilization. And we use an improved datapath where its bit width is configurable throughout the entire SIMD datapath. According to the analysis in Section 2.6, the reduced bit datapath is also able to satisfy the communications requirement. Therefore, bit widths as small as 4 bit—suitable for IoT—is supported and it results in more than a **1.46**× power reduction compared to an 8 bit only SIMD Unit.

3.3.2.2 Scalar Unit

The Scalar Unit is responsible for scalar computations. Bypassing from the SIMD Unit to the Scalar Unit is supported to avoid unnecessary memory accesses. According to use-case analysis, two dedicated ALUs are used in the Scalar Unit shown in Figure 3.7. The Complex Phase Converter [49] is used to convert between complex domain and phase domain while the Magnitude unit is used to extract the magnitude of a complex number. These are very common computations in communications systems and can be clock-gated when not required.

3.3.2.3 Low Power Techniques

To meet the tight power budget, we fully utilize the IoT communications characteristics discussed in Section 2. Several techniques are employed to reduce the power consumption of the proposed IoT SDR datapath. The power reduction of each technique is detailed in Section 5.1 and 5.3.

Voltage Scaling is applied to the entire IoT SDR datapath to reduce the power consumption. The streaming data rate is slow in typical IoT applications, resulting in scenarios in which the required frequency of the IoT SDR datapath is low even if it is fully time-multiplexed. This observation exposes a huge time margin for voltage scaling. Furthermore, in order to voltage scale this design as a single voltage domain, we optimize some sub-designs, such as the Complex Phase Converter, to ensure timing closure.

Clock Gating is applied when some part or the entire IoT SDR datapath will not be utilized for a large time window.

The **IDLE** configuration is used when the IoT SDR datapath is not used for a small time window. As we mentioned in Section 3.2.1, we specialize this design to force the IDLE configuration to eliminate almost all switching activity, resulting in significant power savings.

3.4 Evaluation Method

3.4.1 Signal Processing Kernels

We examined elementary signal processing kernels and combined them to support different end-to-end systems. As we discussed in Section 1.2, kernel functions are very similar in multiple standards. By changing the vector sizes and kernel parameters, ANT and 802.15.4g FSK-SUN can be supported using a similar datapath as Bluetooth Low

Energy (BLE). 802.15.6WBAN, based on Differential Phase Shift Keying (DPSK), has a similar datapath to 802.15.4-OQPSK.

3.4.2 Two Upper Bound Applications

802.15.4-OQPSK and BLE were selected as two upper bound systems to evaluate the IoT SDR architecture, because their bandwidths are highest and they are widely used. Because these two applications represent the upper-bound of computational complexity, it is sufficient to show that the SDR system can achieve many points with lower complexity, under the curve in Figure 3.1. 802.15.4-OQPSK represents standards based on quadrature linear modulation schemes while BLE demonstrates filtered Frequency-Shift Keying (FSK) based schemes. We built the end-to-end system for each, which includes a transmitter, channel, and receiver, in MATLAB to validate the communications datapath and to evaluate the BER. The receiver baseband processing, a combination of hand-coded kernels, was mapped onto the IoT SDR datapath.

802.15.4-OQPSK is the physical layer, defined in the standard, with a maximum data rate of 250kbps (2M signal bandwidth due to spreading spectrum). The modulation scheme is Offset-Quadrature Phase Shift Keying (OQPSK) [50]. The phase of the signal is varied to carry different information; however, OQPSK has continuous phase which yields much lower amplitude fluctuation. The most computationally intensive kernels, including Rate Conversion & Low Pass Filter (LPF), Match Filter and Synchronization of 802.15.4-OQPSK, are all in the complex domain [51]. The communications datapath of 802.15.4-OQPSK receiver is shown in Figure 3.12. The re-sampler is a scalar computation that converts the over-sampling rate to the symbol rate after the correct synchronization information is acquired. The de-spread is a vector operation that correlates the incoming symbol with candidate symbols. The IoT SDR frequency selection and system mapping will be discussed in Section 6.1.

Bluetooth Low Energy (BLE), also marketed as Bluetooth Smart, is designed for low power devices. The digital baseband processing of BLE is the same as Classic Bluetooth and Bluetooth Enhanced Data Rate (EDR). The communications datapath of a BLE receiver is shown in Figure 3.13. This communications datapath is for the maximum BLE data rate of 1Mbps which is also the data rate that Classic Bluetooth supports. The modulation scheme of BLE, Classic Bluetooth and Bluetooth EDR is Gaussian Frequency Shifting Keying (GFSK) [50], in which the information is carried in the frequency domain, meaning that the parts of the most computationally intensive kernels are working in the real domain [52].The over-sampling rate of Bluetooth at 1Mbps is 4M real samples per second.

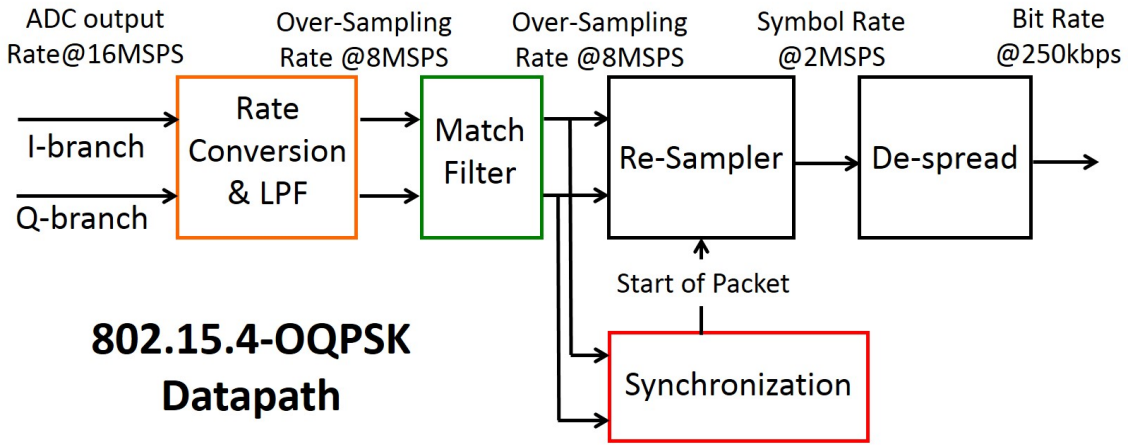


Figure 3.12: **802.15.4-OQPSK Receiver Communications Datapath.** The modulation scheme is **O**ffset-**Q**uadrature **P**hase **S**hift **K**eying. The over-sampling rate is **8MSPS**.

The fact that BLE operates in the 8-bit real domain while 802.15.4 operates in the 4-bit complex domain illustrates the benefit that our multi-level output adder tree and flexible SIMD-ALU design are able to compute both protocols efficiently on the same underlying hardware.

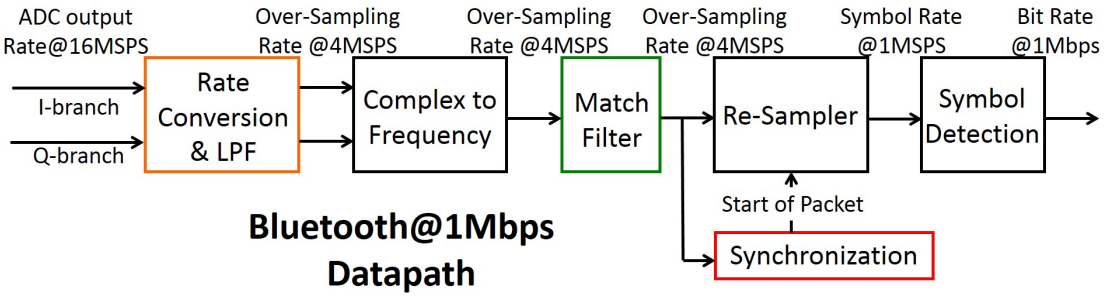


Figure 3.13: **BLE Receiver Communications Datapath.** The modulation scheme is **GFSK**.The over-sampling rate is **4MSPS**

3.4.3 Mapping Representative Applications to IoT SDR Architecture

In this section, we present the methods used to map the receiver communications datapaths of 802.15.4-OQPSK and BLE onto the IoT SDR datapath. Table 3.1 lists the computational characteristics of each kernel in the receiver communications datapaths of 802.15.4-OQPSK and BLE.

Table 3.1: **Computational Characteristics and Kernel Mappings for the two full receiver systems.**

Kernel Name	Vector Size	Data Type	Output Rate	Mapping Unit
802.15.4 OQPSK				
Rate Conversion&LPF (down sample by 2)	32	8-bit/samp complex	8MSPS	SIMD Unit
Matched Filter	8	8-bit/samp complex		SIMD Unit
Synchronization*	128	4-bit/samp complex		SIMD Unit (4b-comp-mult) Scalar Unit (magnitude)
Re-sample	NaN	8-bit/samp complex	2MSPS	Scalar Unit
De-spread	32	8-bit/samp real	1MSPS	SIMD Unit
BLE				
Rate Conversion&LPF (down sample by 4)	32	8-bit/samp complex	4MSPS	SIMD Unit
Complex to Freq	NaN	8-bit/samp complex		Scalar Unit (Complex to Phase)
Matched Filter	16	8-bit/samp real		SIMD Unit
Synchronization*	128	8-bit/samp real		SIMD Unit (subabs) Scalar Unit (magnitude)
Re-sample	NaN	8-bit/samp real	1MSPS	Scalar Unit
Symbol detection	NaN	8-bit/samp real		Scalar Unit
*Synchronization has a longer vector size with 8b/samp that cannot be fit into the 32-lane, 16-bit SIMD Units. We break the computation into two steps, so either two SIMD Units or one SIMD Unit time-multiplexing by two steps can complete the computation of Synchronization.				

3.4.4 Power and Area Estimation Methodology

The methodology to estimate the power and area of the IoT SDR design is presented in Figure 3.14.

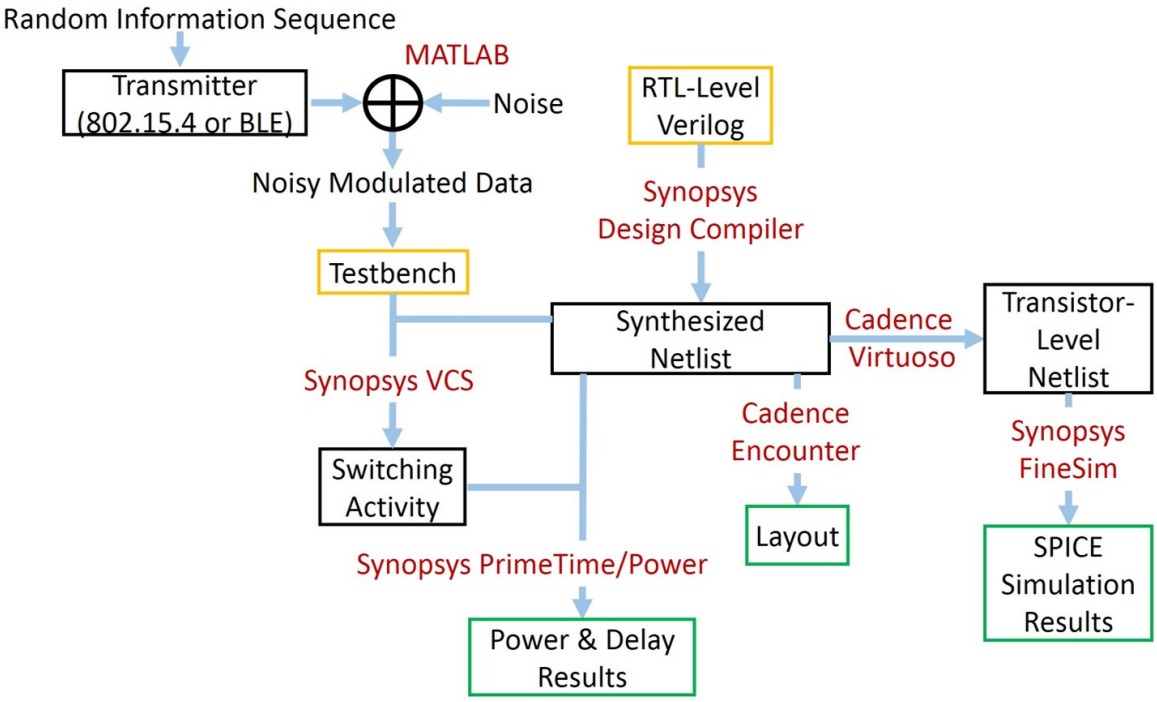


Figure 3.14: Power and Area Estimation Method.

The RTL-level specification Verilog and the testbench, along with noisy modulated data, are used to verify the correctness and estimate the power of the IoT SDR architecture. The worst case delay is acquired from PrimeTime/Power, and the design is placed and routed to obtain the layout for area estimation.

The noisy modulated data is generated as the input to the testbench. First, the information sequence is generated randomly as the input of the 802.15.4-OQPSK or BLE transmitter which is defined in the specifications [2][4]. Second, to simulate the received data sequence, a noisy channel is applied to the modulated data and the output is used as the input of the testbench so as to capture the realistic circuits switching activities.

The IoT SDR design is synthesized in a 28nm technology. The synthesized netlist and the switching activity are used in the Prime Time/Power and SPICE simulations to further estimate the power and the critical path delay. The switching activity captured in this process is close to that obtained with a realistic communication system. As a result, the power and the critical path delay at nominal voltage from Prime Time/Power is realistic.

The same set of noisy modulated data is also fed into the MATLAB full system simu-

lation. The output of the IoT SDR testbench is recorded and compared with the full system result to ensure the functional correctness of the IoT SDR design.

SPICE simulation is conducted to estimate the power and the critical path delay for the voltage scaling. The design is placed and routed using Cadence Encounter in a 28nm technology. The area estimation is based on the final layout.

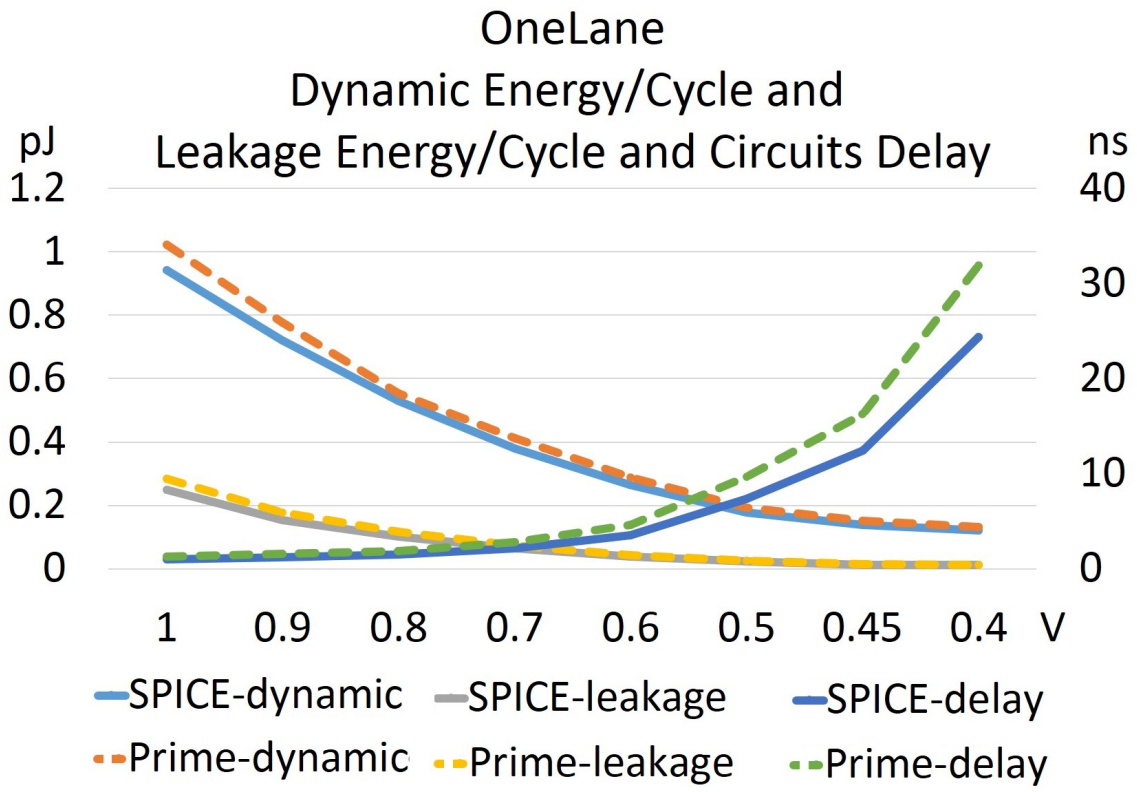


Figure 3.15: **Voltage Scaling Trend of Representative Circuits.** Voltage Scaling is used to meet the power constraint. The Voltage Scaling Ratio is acquired from SPICE.

3.5 Experimental Results

In this section, the power and area results are presented. As we discussed in Section 2.5, the Synchronization overhead cannot be ignored. In this work, the **peak power** of the Synchronization mode is used to characterize the IoT SDR architecture.

3.5.1 Power with Voltage Scaling

We conducted SPICE simulation to get the voltage scaling ratio of this 28nm technology on the representative circuits, and we applied these results to the rest of the datapath.

First, we choose two representative circuits: the OneLane submodule in the SIMD Unit and the Complex Phase Converter submodule in the Scalar Unit to represent the entire design. Second, as shown in Figure 3.15, both SPICE simulation and PrimeTime/Power simulation are applied to the representative circuits at 1V/0.9V/0.8V. Figure 3.15 indicates that the two simulation methods give the same trends on dynamic power over this voltage range. Therefore, we can reliably use the SPICE simulation trends at lower voltage range and apply it to the entire SIMD Unit. Similarly the leakage power and circuits delay also track closely. The same analysis is performed on the Complex Phase Converter submodule and applied to the entire Scalar Unit. Figure 3.16 shows voltage scaling trend of the entire IoT SDR baseband datapath, in terms of energy/cycle and critical path delay.

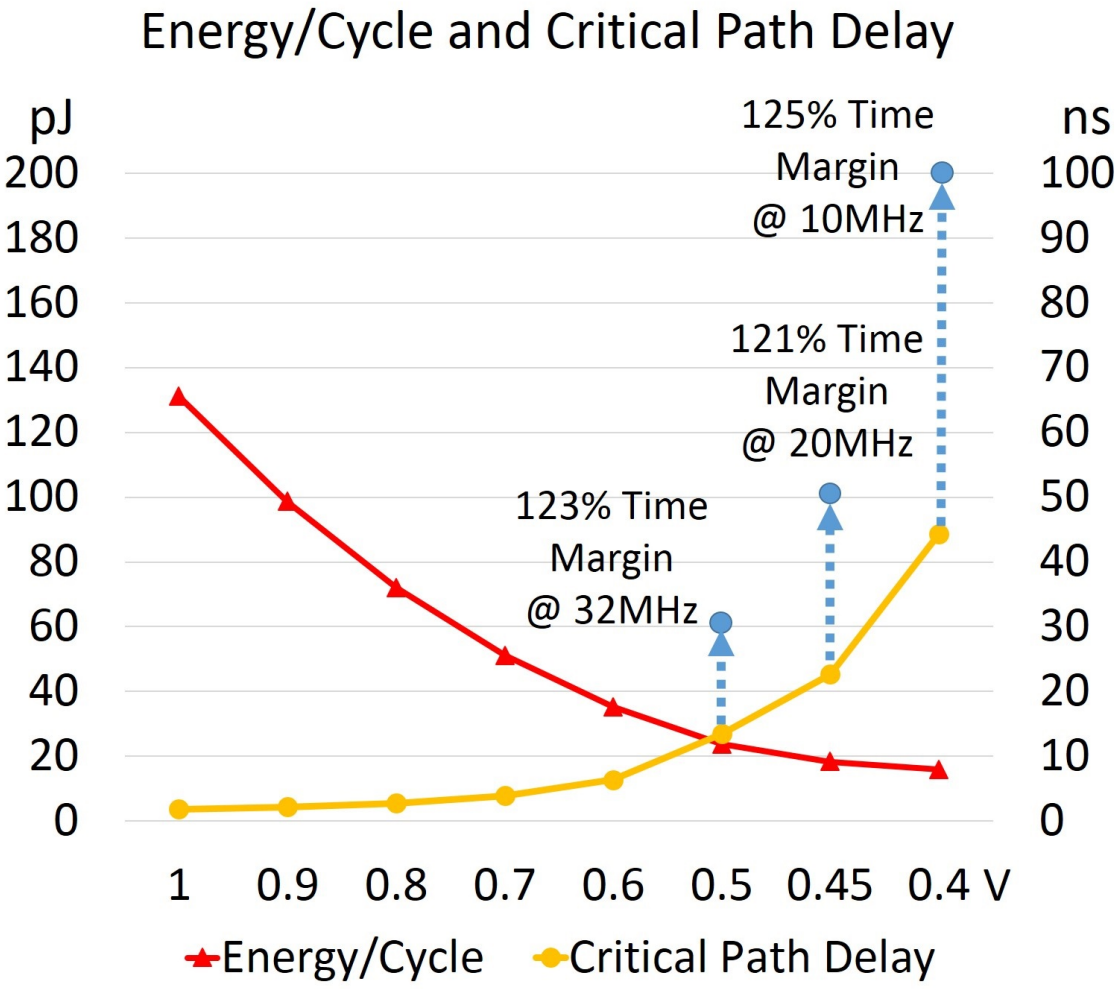


Figure 3.16: **Voltage Scaling Trend of the IoT SDR baseband datapath.** A more than **100%** margin is reserved at each operating points for reliable operation.

Based on the different frequency requirements, we select different operating voltages.

For example, the datapath running at 32MHz operates at 0.5V, the datapath running at 20MHz operates at 0.45V, and below 20MHz the datapath operates at 0.4V. We stop scaling at 0.4V as the threshold voltage for this technology is 0.3V, and the system performance is known to degrade significantly below the threshold voltage [53].

To ensure reliable operations at low voltages, we reserve a more than 100% time margin for critical paths at each operating frequency. Former fabricated chips in 28nm [54] indicate this margin is large enough for reliable functionality. This time margin is equivalent a 50mV voltage margin, largely exceeds the 17mV voltage margin in the near-threshold region suggested by Seo, S. et al. [55] for wide SIMD processors.

The SRAM and MCU are operated at nominal voltage.

3.5.2 Power Breakdown

Figure 3.17 reveals the power breakdown of the IoT SDR datapath. The SIMD-ALU and Register File dominate the power consumption at around 50% and 30%, respectively, at all frequency settings.

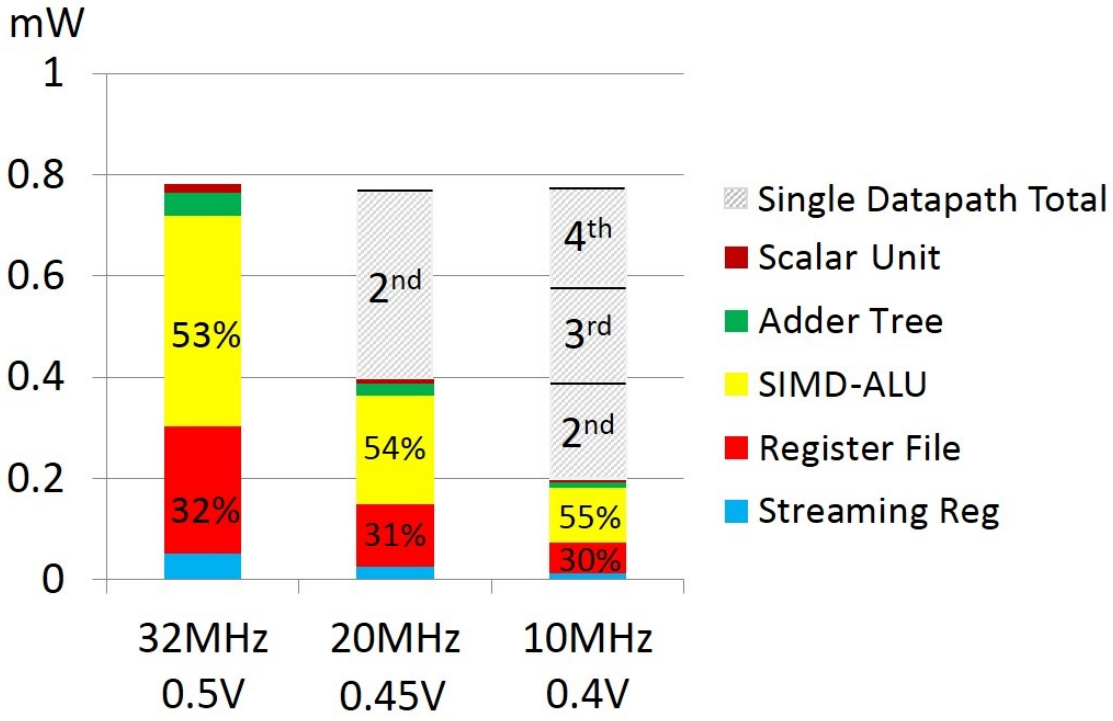


Figure 3.17: **Power Breakdown.** The SIMD-ALU and Register File dominate the power at all frequency settings.

In order to provide the same throughput, more IoT SDR datapaths are required if they

are running at a lower frequency. This is illustrated in Figure 3.17 by stacking two SDR datapaths at 20MHz, and four at 10MHz. The total heights of the stacked datapaths represent approximately the same total throughput. There is neglectable difference in power for the IoT SDR datapath. However, the total system power varies from different system configurations due to memory traffic differences, which will be discussed in Section 6.1.

The energy efficiency for each operating configuration in Figure 3.17 is more than 10Top/J, which is very efficient and comparable to former ASIC designs in 28nm [54].

3.5.3 Power for Different Kernels

We analyze the power for four different operations for the SDR datapath. They are "FIR", which does filtering operations such as LPF and matched filter, "SYNCH", which does the correlation between known signal with receive data, "IDLE" and "Clock Gating".

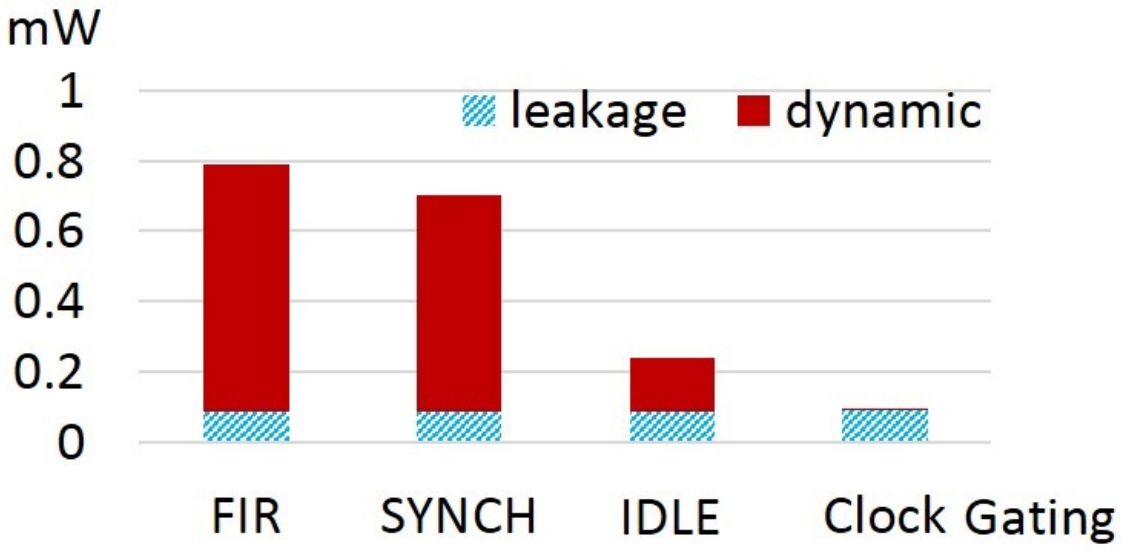


Figure 3.18: Kernel Based Power Consumption.

Figure 3.18 reveals the differences in power when the IoT SDR datapath is running in each of these operations. "FIR" and "SYNCH" are fed with noisy modulated data in the receiver communications datapath, the power difference between "FIR" and "SYNCH" is due to the specific switching activities for each kernel. As we mentioned in the Section 3.2.1, the IDLE configuration forces most of the combinational logic to a non-switching state because the Streaming Register adopts a 0-bit streaming width, resulting in a **3.3**× total power savings. Clock-Gating can reduce the power by **9.1**×. However, clock-gating will be used only in a large window of idle time due to the delay overhead of clock-gating; for a small idle time window, the IDLE will be applied to save power.

3.5.4 Area

The layout of the IoT SDR datapath is shown in Figure 3.19. Its area is 0.074mm^2.

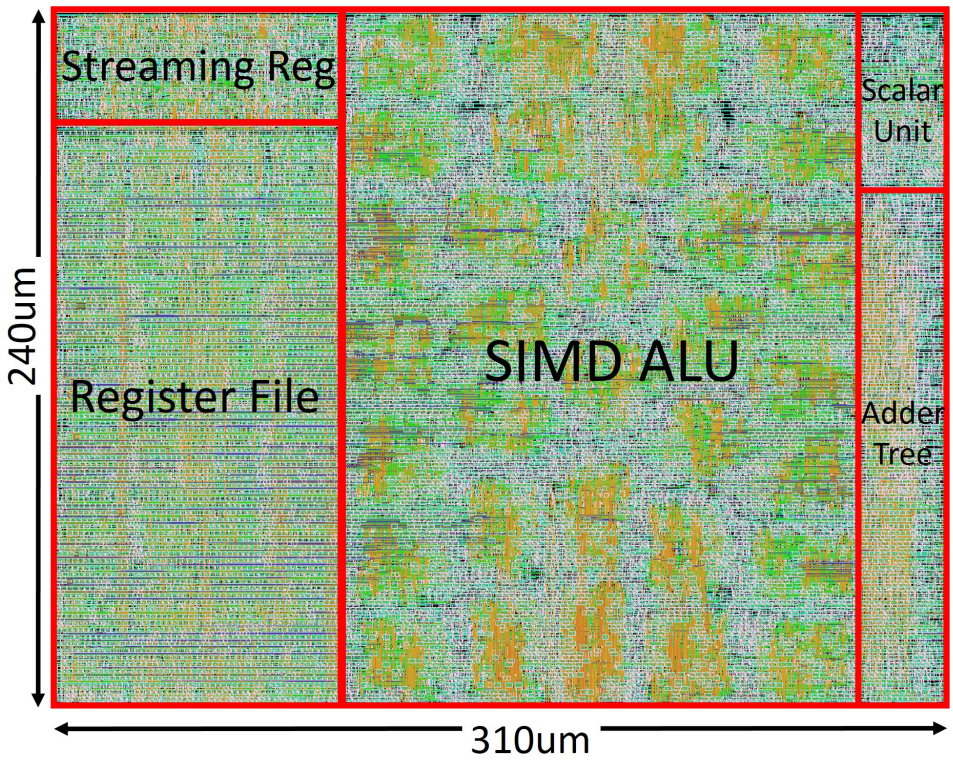


Figure 3.19: Layout

3.6 System Level Trade-off

In this section, the system level architecture of implementing the two full receiver systems, 802.15.4-OQPSK and BLE, onto the IoT SDR architecture is presented.

3.6.1 System Configurations

We evaluate three methods to handle all the kernels for two full receiver systems in real-time: time-multiplexing, duplication of the IoT SDR datapaths, or a combination of the two.

Time-Multiplexing. The IoT SDR datapath requires running at a higher frequency than the over-sampling rate in order to time multiplex across multiple kernels. The required frequency, as listed in Table 3.2, depends on the number of shared kernels and their

Table 3.2: **System Mapping.**

<table>
<tr><th>1× IoT SDR</th><th>2× IoT SDR</th><th>4× IoT SDR</th></tr>
<tr><td colspan="2">Time Multiplexing</td><td>No Time Multiplexing</td></tr>
<tr><td rowspan="4">IoT SDR Datapth
Time Multiplexing by
Rate Conversion & LPF.
Matched Filter,
Synchronization-Step0,
Synchronization-Step1</td><td rowspan="2">IoT SDR Datapath 0
Time Multiplexing by
Rate Conversion & LPF,
Matched Filter</td><td>IoT SDR Datapath 0
Rate Conversion & LPF</td></tr>
<tr><td>IoT SDR Datapath 1
Matched Filter</td></tr>
<tr><td rowspan="2">IoT SDR Datapath 1
Time Multiplexing by
Synchronization-Step0,
Synchronization-Step1</td><td>IoT SDR Datapath 2
Synchronization-Step0</td></tr>
<tr><td>IoT SDR Datapath 3
Synchronization-Step1</td></tr>
<tr><td>Freq >8M + 8M/4 +
2×8M **</td><td>Freq0>8M+8M/4 **
Freq1>8M+8M **</td><td>Freq0>8M **
Freq1>8M/4 **
Freq2>8M **
Freq3>8M **</td></tr>
<tr><td>1× IoT SDR Datapath
@Freq = 32MHz ***</td><td>2× IoT SDR Datapath
@Freq = 20MHz ***</td><td>4× IoT SDR Datapath
@Freq = 10MHz ***</td></tr>
</table>

** For 802.15.4-OQPSK, Over-Sampling Rate=8MSPS
For BLE, Over-Sampling Rate = 4MSPS
*** Frequency margin has been applied.

computational demands. By adopting time-multiplexing, the incoming data is processed in blocks, which results in greater memory traffic. A larger block size requires more memory but less kernel switching. However, the block size is limited by the delay requirements of the different IoT communications standards. For the two demonstrated systems, the tightest turnaround time constraints are 1.92ms for 802.15.4 and 0.31ms for Bluetooth [56]. For the system configurations in Table 3.3, we choose 0.1ms as the block size time in order to meet the aforementioned timing constraints.

Duplication. In this method, the IoT SDR datapaths are duplicated and dedicated to different kernels. Inter-core buf- fers are added to avoid unnecessary memory accesses. For the method, "4× IoT SDR Datapath" in Table 3.2, each datapath is statically scheduled and dedicated to handle one kernel. In this configuration, streaming data is processed sample by sample. Hence, the SDR datapath can run at a frequency close to the over-sampling rate. After the synchronization mode, the SDR datapaths, which were dedicated to do synchronization, can be either scheduled to handle other vector computations or can be clock gated in the case of no active tasks.

Combining Time-Multiplexing and Duplication. In Table 3.2, the "2× IoT SDR Datapath" represents the combination of time-multiplexing and duplication. The SDR datapaths are duplicated and also time multiplexed by two tasks. Since each SDR datapath is

Table 3.3: **Receiver System Configurations. MCU and Memory operate at nominal voltage.**

Configuration					
802.15.4 OQPSK	SDR IoT Datapath		MCU	Memory	
	Frequency	Supply Voltage		Size	Supply Voltage
1× SDR IoT Datapath	32MHz	0.5V	Frequency: 96MHz Supply Voltage: 0.9V	16KByte	0.9V
2× SDR IoT Datapath	20MHz	0.45V			
4× SDR IoT Datapath	10MHz *	0.4V		8KByte	
BLE	SDR IoT Datapath		MCU	Memory	
	Frequency	Supply Voltage		Size	Supply Voltage
1× SDR IoT Datapath	20MHz	0.45V	Frequency: 96MHz Supply Voltage: 0.9V	16KByte	0.9V
2× SDR IoT Datapath	10MHz	0.4V			
4× SDR IoT Datapath	5MHz *			8KByte	

* Frequency(IoT Datapath) must be greater than Sample Frequency

** No memory for IoT Datapath, memory is only for M0+

shared by fewer tasks compared to the single time-multiplexing datapath, the SDR datapath is able to run at a lower frequency. It is possible to scale down to a lower voltage. Since the streaming data is processed block by block, support from memory is still necessary.

However, "2× IoT SDR Datapath" listed in Table 3.2 is not the only mapping policy. For example, we can also dedicate the first SDR datapath to "Rate conversion&LPF" while the second SDR datapath running at 20MHz can still handle the other three kernels. In this mapping, there is no need to buffer the raw ADC outputs. This can bring a big saving in terms of memory if the ADC output rate is much higher than the over-sampling rate.

For the system configuration with duplicated IoT SDR datapaths, an inter-core buffer between the IoT SDR datapaths is added to pass values between kernels. The memory interface and inter-core buffer overhead is fairly small on top of the stacked datapaths in Figure 3.17. The memory interface is less 1% in all three mapping policies and the inter-core buffer takes less than 3% in combination and duplication methods. Because the nature of the incoming data is streaming, both the memory access and intermediate buffering fall into a very fixed pattern. The IoT SDR datapath only writes correlation results from the synchronization kernel to the memory and does not read values. The rest of the memory is used almost exclusively by the MCU.

So far, we have described the design of only the IoT SDR datapath. In the following paragraph, we explore other components, particularly the MCU and the memory.

For the memory, we use a 45nm SRAM compiler. The area of the memory is scaled down to the equivalent size in a 28nm technology according to results of Wu et al. [57], while the power of the memory is scaled down to the equivalent value in a 28nm process [57]. We run the memory at the nominal voltage to avoid any reliability issues [58].

An ARM M0+ running at 96MHz is chosen as the MCU to support the two system benchmarks. The ARM M0+ is responsible for configuring and scheduling the IoT SDR datapath and handling control-related computations. The power and area of the ARM M0+ is derived from [59] and scales from 40nm to 28nm according to [60]. As shown in Table 3.3, only the IoT SDR datapaths are scaled down to a low voltage, the MCU and memory operate at their nominal voltage. The MCU is a hard macro which we cannot perform timing closure on at lower voltages.

3.6.2 System Power and Area

The total system power and area are listed in Table 3.4. The power in this table is obtained by assuming the system is running in synchronization+FIR mode, which is the peak power mode of the system.

Table 3.4: **System Power and Area**

802.15.4 OQPSK	1× IoT SDR		2× IoT SDR		4× IoT SDR	
	Power (mW)	Area (mm^2)	Power (mW)	Area (mm^2)	Power (mW)	Area (mm^2)
IoT SDR Datapath	0.801	0.074	0.825	0.149	0.820	0.298
MEM	0.925	0.024	0.690	0.024	0.308	0.014
MCU	0.265	0.003	0.265	0.003	0.265	0.003
Total	**1.991**	**0.101**	**1.780**	**0.176**	**1.393**	**0.315**

BLE	1× IoT SDR		2× IoT SDR		4× IoT SDR	
	Power (mW)	Area (mm^2)	Power (mW)	Area (mm^2)	Power (mW)	Area (mm^2)
IoT SDR Datapath	0.417	0.074	0.413	0.149	0.487	0.298
MEM	0.690	0.024	0.494	0.024	0.236	0.024
MCU	0.265	0.003	0.265	0.003	0.265	0.003
Total	**1.372**	**0.101**	**1.172**	**0.176**	**0.988**	**0.315**

Figure 3.20 illustrates the power/area trade-off for three system configurations. The system power and area are presented for 802.15.4-OQPSK. The analysis also holds for BLE. Considering the scheduling flexibility, the ability to react to complicated practical scenarios such as the high ADC output case discussed in Section 6.1 and the scalability to handle future higher data rate applications, the duplication methods, "2× IoT SDR Datapath" and "4× IoT SDR Datapath" expose more flexibility and scalability, with the penalty of area.

3.6.3 System Discussion

The proposed IoT SDR baseband system demonstrates a power consumption less than 2mW for two upper bound applications at every configuration, with the combination of all architectural enhancements and low power techniques. As detailed in Section 3.2, the main architectural enhancements—streaming registers, dedicated complex/real ALUs as well as ultra small 4-bit operations—contribute in power reduction of $10.4\times$, $4.7\times$ and $1.46\times$ individually, resulting in a total of a $71\times$ difference. Without these architectural enhancements, the SDR baseband processor power consumption will grow up to more than 140mW, which is unacceptable for IoT devices.

Our proposed IoT SDR system, which can support many standard or nonstandard IoT communications, is compared with the best commercial solution, Nordic nRF8001 [61].

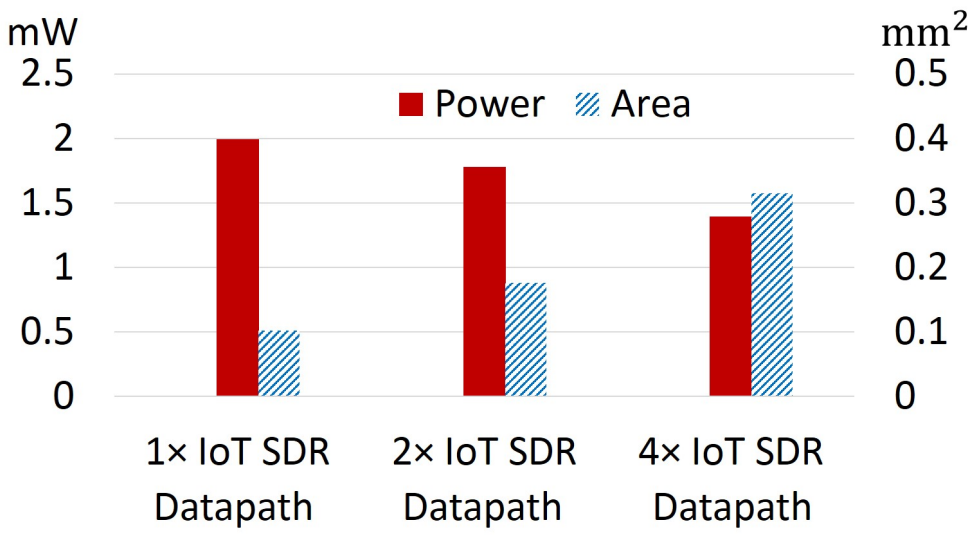


Figure 3.20: System Power and Area Trade-offs.

Nordic is a highly integrated single-chip Bluetooth Smart Connectivity IC including RF/analog and digital modules. Its power consumption is 23.8mW. The peak power of our configurable digital system is at most 8.4% of their total power. To our knowledge there is no reference that explicitly breaks out the power consumption for the digital portion of IoT radio devices. Prototypes, typified by [62][63][64], focus on RF/analog and only include small portion of digital baseband processing. Sensor network research [65][66][67] targets for topics such as network layer processing, even they measure the power/energy consumption of the sensor nodes, utilizing commercial transceivers.

3.7 Related Work

Previous SDR work, such as SODA [68], AnySP [69] and Ardbeg [70], have focused on wireless standards for mobile handheld platforms, typically LTE type wide bandwidth communications with hundreds of mW power budget. Our solution differs in that we look at IoT scale devices, which have two orders of magnitude smaller power constraints. In the IoT space, we identify a different solution. Specifically, our MCU directed baseband processor uses variable bit precision and dedicated complex/real ALUs, a more flexible reduction network, and streaming registers to marshal incoming data. We explored the low sampling rate of IoT communication by employing voltage scaling to reduce the power consumption while maintain more than 100% time margin to ensure reliability.

Our configurable baseband processor, which is controlled by an MCU, is closer to an

ASIC than other SDR designs such as SODA. In this study, we show that the SIMD model can be power scaled to sub mW (with system power less than 2mW). We believe this unique work is pioneering in demonstrating baseband SDR is possible even in a low power IoT domain.

3.8 Conclusion

In this chapter, we present a configurable SDR baseband architecture for IoT devices. Several reasons such as the multi-standard scenario in the IoT application domain presents a promising future for SDR. Our IoT SDR baseband processor supports flexible bit-width, vector-reduction operations and efficient streaming computation, while reducing power consumption by employing several techniques such as voltage scaling and clock gating. This combination enables a $71\times$ power reduction over a more general purpose SDR design.

A comprehensive evaluation on power and area is conducted. The single IoT SDR datapath is placed and routed to fit an area of 0.074mm^2 within mW power consumption. Two representative upper bound systems, 802.15.4-OQPSK and BLE, are mapped on this IoT SDR architecture. The total system power and area trade-off is investigated. The peak power of this proposed digital baseband systems including baseband datapath, MCU and memory, is at most 1.99mW for 802.15.4-OQPSK and 1.37mW for BLE within an area of 0.101mm^2 in 28nm technology.

CHAPTER 4

A Programmable Galois Field Processor for the Internet of Things

This work investigates the feasibility of a unified processor architecture to enable error coding flexibility and secure communication in low power Internet of Things (IoT) wireless networks. Error coding flexibility for wireless communication allows IoT applications to exploit the large tradeoff space in data rate, link distance and energy-efficiency. As a solution, we present a light-weight Galois Field (GF) processor to enable energy-efficient block coding and symmetric/asymmetric cryptography kernel processing for a wide range of GF sizes (2^m, $m = 2, 3, ..., 233$) and arbitrary irreducible polynomials. Program directed connections among primitive GF arithmetic units enable dynamically configured parallelism to efficiently perform either four-way SIMD 5- to 8-bit GF operations, including multiplicative inverse, or a long bit-width (e.g., 32-bit) GF product in a single cycle. To illustrate our ideas, we synthesized our GF processor in a 28nm technology. Compared to a baseline software implementation optimized for a general purpose ARM M0+ processor, our processor exhibits a $5 - 20\times$ speedup for a range of error correction codes and symmetric/asymmetric cryptography applications. Additionally, our proposed GF processor consumes $431\mu W$ at $0.9V$ and $100MHz$, and achieves $35.5pJ/b$ energy efficiency while executing AES operations at $12.2Mbps$. We achieve this within an area of $0.01mm^2$.

4.1 Introduction

Envision the future Internet of Things. We envision a future world with a Trillion of IoT devices communicating with other (often heterogeneous) devices. The devices can be configured to address many standards, non-standard, custom specified communication schemes for various use-case scenarios. The communication between things will be enabled at anytime, anywhere among anything, from the Internet of Things (IoT) toward Internet of Everything (IoE).

Today, IoT networks are often fragmented, having nodes with different physical layer protocols such as Bluetooth Low Energy (BLE) [4] or IEEE802.15.4 [2]. In the future, direct connectivity should no longer be limited by a single protocol or one specific physical layer ASIC.

The value of flexible baseband processing has been demonstrated in previous works [68, 70, 71]. By dynamically changing configurations (signal bandwidth, modulation parameters, etc.) on the Software-Define Radio (SDR) platform, IoT devices can adjust the data rate to extend the operating distance with the same power budget. SDR can enable higher energy efficiency when the operating scenarios require shorter distance and/or lower data rates than that are defined in a particular wireless standard. Rapid innovation is enabled by the nature of SDR as new communication modulation schemes and protocols can be easily adopted with a simple software update. Enhanced spectral efficiency can be achieved via dynamic adaptations in frequency planning, pulse shaping, and bandwidth allocation.

Flexible radio solution is realizable without increasing the overall power consumption [71]. In fact, better energy efficiency can be achieved via operating the SDR at an energy-optimal configuration that might be outside the scope of a specific physical layer protocol [72, 73]. In this chapter, we augment the benefit obtained from SDR baseband signal processing flexibility by also adding flexibility to the information / error-correction coding in IoT communication.

4.1.1 Error Correction Coding Flexibility

Error correction coding is a very effective way to enhance reliability in IoT communications. The optimal energy efficiency, data rate, and link distance tradeoff can be obtained by adjusting the error correction coding rate and/or the information encoding schemes. There are many error corrections codes that have been proposed for low power wireless. Today, each physical layer protocol specifies an error correction scheme that is optimized for a representative scenario (combination of packet length, data rate and link distance). For example, Bose-Chaudhuri-Hocquenghem (BCH) coding is the error correction coding method used in Body Area Networks [3], that allows 2 or 3-bit error correction in a 63-bit codeword. Reed Solomon (RS) codes also have been proposed for low power communication because they work within the hardware complexity restrictions for Low Data Rate Wireless Personal Networks (LDR WPAN) [74, 75].

A single fixed error correction code is suboptimal, given that baseband processing is migrating towards a more versatile SDR with flexible data rates, bandwidths and modulation schemes to satisfy heterogeneous IoT operating scenarios. The information coding

flexibility, therefore, plays an important role in data-rate, distance, and energy-efficiency tradeoffs for IoT communications. Ideally, flexible coding should support a broad class of parameterized coding schemes for various standards or non-standard communications to adapt to different channel conditions. A low rate encoding with strong error correction capability should be applied in a noisy channel, while in a clean channel we should be able to pack more information bits in a code word. A flexible coding scheme should also be able to address different error patterns, for example, be robust enough to handle both uniformly distributed and burst bit errors. Unlike convolutional, turbo, LDPC and polar codes that are widely used in high performance broadband access, low power IoT communications, with relatively low data rates and short packet lengths, do not employ a long codeword. In this work, we address a broad class of Galois Field (GF) based block coding schemes with a short ($< 100s$ bits) codeword as they are more common for low power IoT wireless connectivity. By varying codeword length n, information length k and GF size (2^m), various block codes are supported that can optimally adapt to different channel conditions and application scenarios.

4.1.2 Cryptography

While the proposed flexible error correction coding schemes provide robustness against noise and interference, IoT wireless communication that exchanges a plaintext through a shared medium is fundamentally insecure to malicious attacks that eavesdrop and impersonate in the middle of the network. Our proposed GF processor addresses the security aspect of the IoT communication by providing a unified architecture to perform not only error correction coding but also popular cryptography kernel functions computed in a finite GF.

Asymmetric Cryptography. Asymmetric cryptography such as Elliptic Curve Cryptography[1] [76], is widely employed in an authentication process to exchange a shared private key. However, because of its extremely large GF (e.g., 2^{233}), ECC$_l$ complexity is orders of magnitude higher than that of the symmetric cryptography process. Although, the asymmetric cryptography process is called only once when a new secure session is established, ECC$_l$ software implementation to perform long bit length GF operations is very inefficient and typically incurs excessive latency when realized on a low power general purpose processors (analysis provided in Section 4.3.3.4). Because it is used infrequently, the area of an ECC$_l$ solution is more critical than its energy efficiency provided it meets the latency requirement. Our proposed GF processor architecture provides an area efficient

[1] In this chapter, we use ECC$_l$ to refer Elliptic Curve Cryptography and ECC$_r$ to refer Error Correction Code.

solution for ECC without incurring a significant overhead to the baseline architecture in order to support both ECC_r and ECC_l.

Symmetric Cryptography. Once a secure session is established through an ECC_l based key exchange process, symmetric cryptography using a shared private key is applied on a packet-by-packet basis to encrypt / decrypt a plaintext / ciphertext message pair. A widely used method for symmetric cryptography is the Advanced Encryption Standard (AES). Unlike asymmetric cryptography ECC_l, throughput of the symmetric cryptography should be matched to the data rate of the underlying physical layer.

Traditionally, the coding and security modules are implemented as dedicated accelerators [77, 78, 79] because they are computationally intensive and do not efficiently map onto a general purpose processor architecture. Moreover, due to lack of programmability in the legacy physical layer, coding flexibility has not been widely explored. With the move towards SDR, this opens a new research possibility on microarchitectures to enable coding flexibility and to inherently address the security aspect of IoT connectivity as well.

4.1.3 Feasibility to Address Coding Flexibility and Cryptography on the Same Piece of Hardware

Coding flexibility allows IoT applications to exploit the large tradeoff space in data rate, link distance and energy-efficiency. However, realizing coding flexibility on a general purpose processor is too inefficient, and it easily leads to 100× worse energy efficiency compared to dedicated hardware accelerators, offsetting the benefit of flexible coding. On the other hand, employing multiple dedicated accelerators is expensive in terms of design effort and production cost. Addressing this challenge, we propose a unified architecture that is area- and energy-efficient to support not only flexible information coding but also secure wireless communication via asymmetric and symmetric cryptography.

To demonstrate feasibility of the proposed solution, we first briefly describe the datapath involved in example error correction coding and cryptography processes. Fig. 4.1(a) show the datapath of the binary BCH code [80, 81], which has three coding parameters (n, k, t); n represents the codeword bit length, k represents the information bit length and t indicates the number of bit errors correctable within a codeword. A binary BCH code (n, k, t) is constructed from a Galois field GF(2^m), where $n = 2^m$. The decoding datapath of binary BCH consists of four kernels. The first kernel – Syndrome Calculation – evaluate the received codeword. Non-zero syndromes indicate that errors exist in the received codeword. The kernel – Closed Form ELP [82] – solves the coefficients of the error locator polynomial. Finally, the Chien search finds the roots of the error locator polynomial, and

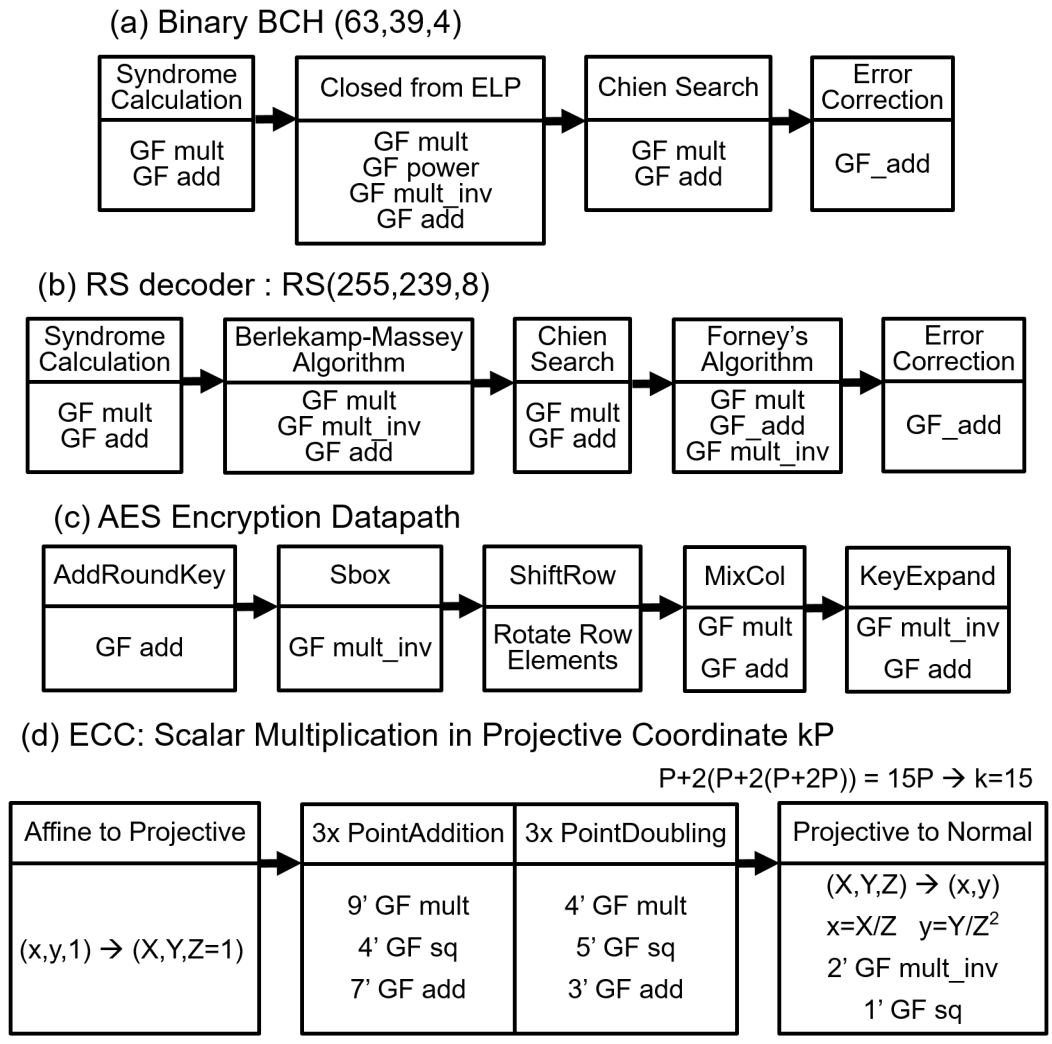


Figure 4.1: Dataflow for ECC_r, AES, and ECC_l. They share the same underlying GF arithmetic operations.

the roots reveal the error locations. Bit errors at these locations are corrected via GF additions. The BCH code is favored by many low power devices [83, 84, 85] for its concise decoding process and effectiveness in correcting uniformly distributed random errors.

Reed Solomon (RS) codes [80, 86] also have (n, k, t) parameters and operate on GF(2^m), where $n = 2^m$. Unlike binary BCH, in an RS code, n is the number of symbols in a codeword, where each symbol has m bits. k and t are the number of information symbols and the number of correctable error symbols in a codeword, respectively. The decoding datapath is very similar to binary BCH. The Berlekamp-Massey-Algorithm(BMA) is a generalized method to solve the coefficients of the error location polynomial. RS decoding requires an additional kernel—Forney Algorithm—to evaluate the actual m-bit error values after the error symbols are located via the Chien search algorithm. RS codes are suitable for

correcting multiple-burst bit errors (up to m error bits within a symbol). A generalized RS decoder datapath is shown in Fig. 4.1(b).

A variety of BCH and RS codes can be created with different coding rates (k/n), GF sizes (2^m) and the error correcting ability represented by the parameter t. As Fig. 4.1(a) and (b) indicates, BCH and RS codes share an identical datapath at a high level. One of technical challenges involved in flexible block coding is efficient handling of various GF sizes (2^m) and irreducible polynomials associated with each GF [87]. Dedicated hardware accelerators address this issue by designing a hard-wired GF arithmetic unit that is specific to a given GF size and an irreducible polynomial pair.

The datapaths of AES and ECC$_l$ shown in Fig. 4.1(c) and (d) respectively, which are distinct from that of the error correction codes. Our proposed architecture exploits the fact that each step involved in AES and ECC$_l$ can be converted to equivalent finite Galois field arithmetic operations. For example, the S-box operation in AES is equivalent to GF(2^8) inverse operation [88]. Similarly, the point addition operation in ECC$_l$ can be mapped to a series of GF multiplications, additions, and division in GF(2^m), where typically $m > 100$.

Fig. 4.1(c) illustrates the AES encryption datapath for a single round of processing. Different key lengths (128, 192 or 256-bit) determine the number of rounds for a single encryption. Decryption is very similar to encryption except that the kernels process in a reverse direction with different constants to implement inverse.

The ECC$_l$ datapath is shown in Fig. 4.1(d). The key operation in ECC$_l$ is scalar multiplication (kP) performed on an elliptic curve [89, 90], where k is the scalar and $P(x_p, y_p)$ is a point on the elliptic curve. The scalar multiplication is further decomposed into two operations–point addition (P_1+P_2) and point doubling $(2P)$. Both point addition and point doubling are computed via several GF multiplication, square and multiplicative inverse operations. Depending on the ratio between multiplication and multiplicative inverse, transforming points to a different coordinate (e.g., the projective coordinate) may be necessary to reduce the complexity.

We make an important observation that all coding and cryptography processes under consideration (binary BCH, RS, AES and ECC$_l$) were derived from GF mathematics, therefore they share the same underlying operations. However, their GF sizes and irreducible polynomials can be very different. Addressing this challenge leads to our proposed GF processor architecture described in Section 4.2. We propose to implement the Galois field arithmetic unit as a dedicated GF functional unit of the proposed processor to provide efficient computation of complicated GF operations.

4.1.4 Contribution

To address generality in both coding and cryptography, we need to support arithmetic flexibility in different bit-widths and arbitrary irreducible polynomials associated with each Galois field. Meanwhile, the cost of arithmetic flexibility needs to be affordable because IoT devices are limited by strict power budget and area constraints. The contributions of this work are summarized as follows:

- We analyze several error correction coding schemes and cryptography kernels to identify the shared fundamental operations.
- We propose a flexible bit-width Galois field arithmetic unit consisting of two types of primitive arithmetic units: multiplication and square.
- These primitives are combined to support single cycle complicated instructions: multiplicative inverses and long bit-width GF products using 32-bit segmentation.
- We exploit the parallelism in coding and cryptography datapaths, and employ (configurable) four-way 8-bit SIMD operations to improve performance and functional unit reuse. The SIMD computation datapath is shared among the multiplicative inverse and long bit-width GF product instructions.
- We illustrate how to efficiently handle extremely long bit-width (> 100bit) multiplication by iteratively using the SIMD-optimized single-cycle 32-bit products.
- The GF arithmetic unit is integrated into a two-stage in-order processor. Several coding kernels and cryptography kernels are evaluated using the proposed processor architecture.

We demonstrate the feasibility of a unified processor architecture to enable coding flexibility and secure communication in low power IoT wireless networks. The proposed processor enhances wireless communication energy-efficiency by adapting the coding scheme to various noise/interference conditions. At the same time, secure communication is realized via symmetric and asymmetric cryptography processes that are also efficiently implemented on the proposed processor.

4.2 The Programmable Light Weight Galois Field Processor

4.2.1 Processor Architecture

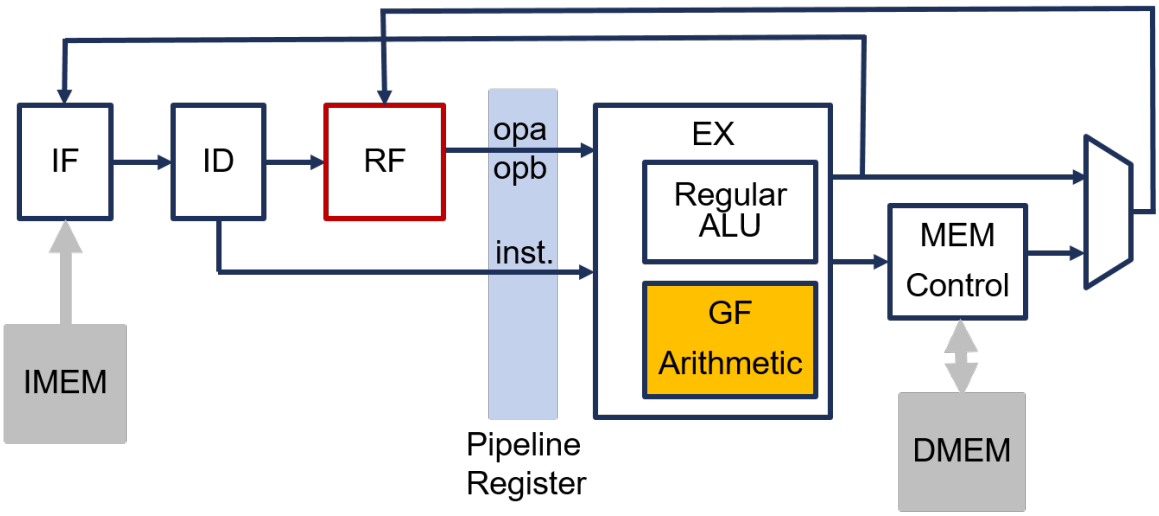


Figure 4.2: Proposed Galois Field Processor

The system architecture of the programmable Galois field processor is illustrated in Fig. 4.2. Control related computation, integer arithmetic operations and memory operations are conducted by regular functional units. Rather than implementing the full instruction set of a Cortex M0+, we profile the workloads and identify the subset of control instructions, integer arithmetic operations and memory operations needed. We only implement this subset to conserve area. We use a 32-bit datapath, and 16 entry 32-bit register file. We add a Galois field arithmetic unit and associated instructions to optimize GF operations. All SIMD GF instructions, including multiplication, square/power, multiplicative inverse, and a 32-bit partial product are single cycle instructions. Arbitrary irreducible polynomials with degree smaller or equal to 8-bit are supported with the dedicated configuration register (inside the GF Arithmetic Unit). The microarchitecture of GF arithmetic unit is discussed in section 4.2.2 - 4.2.4.

4.2.1.1 Instruction Set Architecture

Our instruction set architecture (ISA) is a combination of GF instructions and a subset of Cortex M0+ instructions, which conduct all non-GF instructions including control-related, memory and integer arithmetic operations. The GF instructions have a 26-bit format, con-

Table 4.1: **Galois Field Instructions.**

Category	Format	Description	Operations
SIMD Instr. Each RF holds four 8-bit GF values. All operations are in GF filed.	gfMult_simd R_{s1},R_{s2}, R_d	Multiplication	$R_{s1} \bigotimes R_{s2} \rightarrow R_d$
	gfMultInv_simd R_s, R_d	Multiplication inverse	$R_s^{-1} \rightarrow R_d$
	gfSq_simd R_s, R_d	Square	$R_s^2 \rightarrow R_d$
	gfPower_simd R_{s1},R_{s2}, R_d	Power	$R_{s1}^{R_{s2}} \rightarrow R_d$
	gfAdd_simd R_{s1},R_{s2}, R_d	Addition	$R_{s1} \bigoplus R_{s2} \rightarrow R_d$
Partial Product	gf32bMult R_{s1},R_{s2}, R_d^h,R_d^l	32-bit carryfree multiplier	$R_{s1} \times R_{s2} \rightarrow (R_d^h, R_d^l)$
Configuration	gfConfig #Address	Load 56-bit coef. to field config. register	$*(Address) \rightarrow R_{config}$

sisting of a 10-bit opcode and a 16-bit register field. Their operation is illustrated in Table 4.1. The GF arithmetic instructions support a complete set of GF arithmetic operations.

4.2.2 The Galois Field Arithmetic Unit

In this design, we restrict our design to a binary Galois field (2^m), where m is the bit width to represent an element in the Galois field. In this arithmetic, m-bit inputs produce m-bit output without a carry. Addition and subtraction are identical; they are implemented via bitwise exclusive-OR. Multiplication is performed as multiplication modulo an irreducible polynomial. Since a GF(2^m) can have many irreducible polynomials, the selection of the irreducible polynomial will affect the multiplication/multiplicative inverse operations. Flexibility in GF arithmetic implies that we need to address various bit-widths and different irreducible polynomials.

4.2.3 Design Challenges

In this section, we identify design challenges that are specific for a flexible Galois Field arithmetic unit.

Most of the error correction codes and AES only require small bit-widths to represent a GF element, specifically $m \leq 8$ as shown on the left side of Fig. 4.3. In contrast, asymmetric cryptography, ECC$_l$ on the right side of Fig. 4.3, requires very long bit-width fields. Several standard binary curves are recommended for ECC$_l$ [91, 92], the smallest being 113 bits and the largest being 571 bits. Although these two requirements differ widely in bit width range, we show that they can be implemented efficiently with the same underlying hardware.

Challenges arise because of the fact that, in contrast to integer arithmetic operations, a smaller GF bit-width ($m \leq 8$) operation cannot directly use a larger GF bit-width datapath by simply setting the most significant bits to zeros. Galois fields with distinct sizes have completely different irreducible polynomials in terms of both degrees and coefficients. Setting the most significant bits to zeros does not work even if the smaller bit-width irreducible polynomial may be a subset of the larger bit widths irreducible polynomial. For example, BCH(31,26,1) on GF(2^5) is a subset of RS(255,239,8) on GF(2^8), as shown in Fig. 4.3. However, even in such cases, setting the most significant 3 bits to zero will not work because Galois field requires a polynomial modulo operation. Fig.4.5 (b) will illustrate this with an example.

Even the same bit-width datapaths cannot directly share the same multiplication and multiplicative inverse units. For example, RS(255,239,8) and AES both in GF(2^8) cannot use the same multiplication and multiplicative inverse units becasue of the different coefficients in their irreducible polynomials. All in all, specific hardware has to be designed to support variable bit-width GF operations.

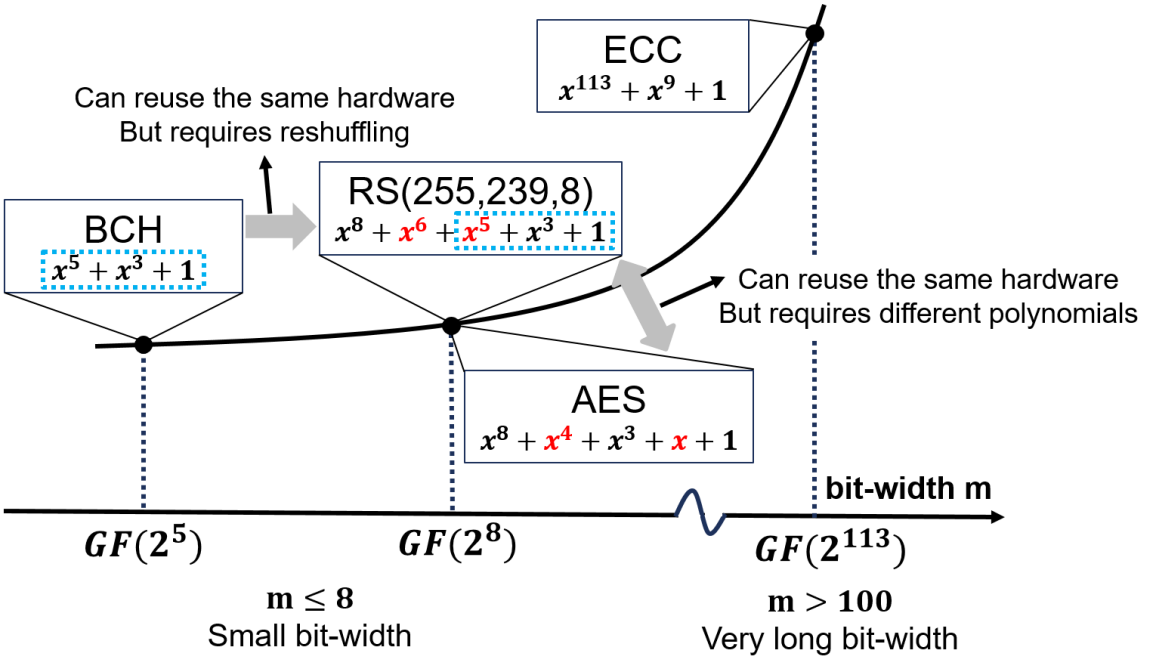


Figure 4.3: GF Arithmetic Design Challenge. To support a wide range of bitwidths and arbitrary polynomials.

4.2.4 The GF Arithmetic Unit Microarchitecture

The Galois field arithmetic unit microarchitecture is illustrated in Figure 4.4. It is composed of several basic units, including 16 8-bit GF multiplication units and 28 8-bit GF square units. It also includes one dedicated configuration register to support arbitrary irreducible polynomial and control logic to selectively connect basic units to execute different instructions.

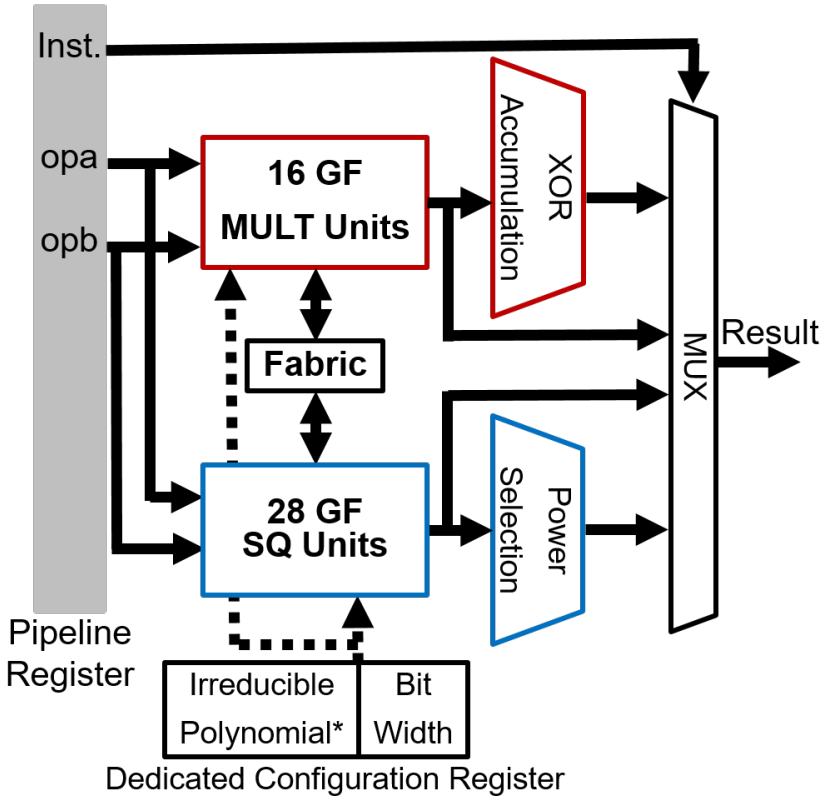


Figure 4.4: Galois Field Arithmetic Unit. It supports SIMD instructions: multiplication, square, and multiplicative inverse, as well as 32-bit partial products. It has a dedicated configuration register shared among the ALUs. The pipeline register is shared between GF arithmetic units and the regular arithmetic logic units.

4.2.4.1 Primitive Units

In this design, we employ two primitive computation units: multiplication and square.

Multiplication. We employ the compact GF multiplier method introduced in [93]. It decomposes multiplication into a carryless multiplier and a polynomial reduction module. The multiplier first computes a $(2m-1)$ bit full product in GF(2) from two m-bit inputs, Fig. 4.5 illustrates this. The polynomial reduction module performs $modulo(\mathbf{c}, \mathbf{r})$, where

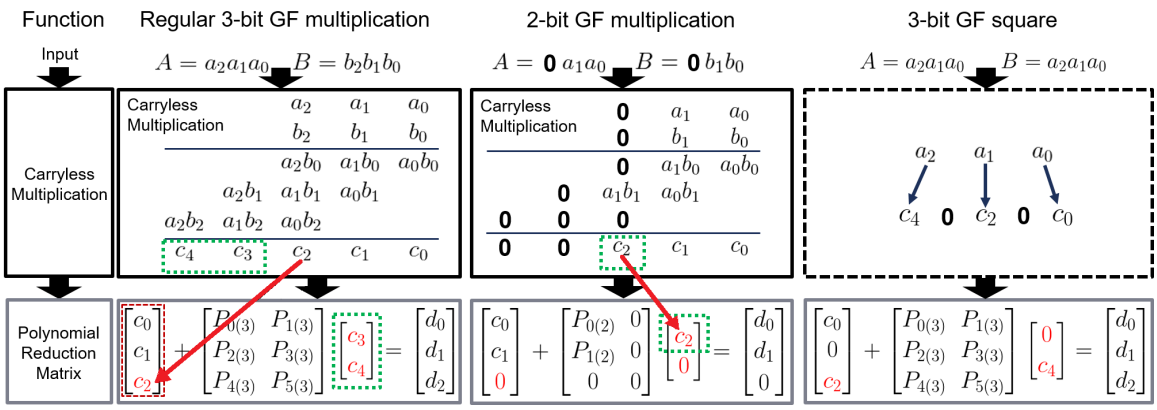


Figure 4.5: GF Multiplication Unit. Illustrating regular multiplication, smaller bitwidth multiplication and squaring.

c is the carryless multiplier output and **r** is the vector representation of the irreducible polynomial.

The polynomial reduction module is implemented as a linear transformation by implementing a GF(2) matrix vector multiplication. The reduction is done using a reduction matrix (**P**), which is derived by a transformation of the irreducible polynomial (**r** $\rightarrow$ **P**), and can be determined a priori. The reduction matrix is programmable by writing the values of **P** to the dedicated configuration register.

The default Galois field operation in our datapath is GF(2^8)–an 8-bit datapath. The polynomial reduction circuits include an 8-by-7 matrix and a reduction vector of 7-bits as outlined by the green dashed box. The remaining vector shown by the red dashed box fills out the datapath bit width.

We deploy a new method, which selectively maps the full product **c** to the polynomial reduction module based on a GF size dependent pattern. The bit-width signal from the configuration register directs the mapping circuit to split the full product **c** into the reduction vector (green dash box in Fig. 4.5) and the remaining vector (red dashed box in Fig. 4.5).

The example of operating on a smaller bitwidth is illustrated in 4.5 (b). This is done by first setting the most significant bits to zeros and then changing the mapping of **c** to the polynomial reduction step. The mapping of **c** depends on the GF size, with each size having a unique mapping. As mentioned in Section 4.2.3, if only the significant bits were set to zero without updating the mapping, the c_2 bit in the partial product would be mapped to the wrong position. Our hardware contains a configuration register which specifies how the mapping of partial products to the polynomial reduction should occur, enabling us to support smaller bit-widths. As a result, we reuse the same size computation resource in the polynomial reduction module. The control overhead to support 5-, 6-, and 7-bit using the

Table 4.2: Resource Comparison for Different Multiplication Methods

		Systolic	This work
Polynomial Modulo		Bit-pipelined	Single Step Linear Transform
Comb.	AND	$2m^2$	$2m^2 - m$
	XOR	$2m^2$	$2m^2 - 3m + 1$
FF	opa	$(m-1)m$	Pure combinational logic
	opb	$(m-1)m/2$	
	intermediate	$(m-1)m$	
Total area*		$16.5m^2 - 10m$	$6.5m^2 - 7.75m$
Configuration Datapath (shared by multiple ALUs)			
FF	p or S	m	$m(m-1)$
*AND : MUX : XOR : FF = 1 : 2.25 : 2.25 : 4 in a 28nm technology			

8-bit computation unit is small: 8% of the entire arithmetic units.

Our method is in contrast to prior approaches for supporting smaller bit-widths [93, 94], which instead increase the programmable part of the reduction matrix-vector operation. An increased size of the reduction matrix would require more computation resources. To support 5-, 6-, and 7-bit multiplication, one option [94] is to add a 5-by-3 matrix vector operation, incurring more than 26% additional hardware overhead. The other option [93] is to include a triangular matrix, which requires more control when mapping the full product to the linear transformation and several additional computations.

We compare our implementation with another configurable GF multiplication — the popular systolic solution [95, 96, 97]. The multiplication resource comparison with a LSB method in [95] is shown in Table. 4.2.

The difference in combinational logic between the two methods is small, because it is determined mostly by the computational complexity. The systolic method processes one bit polynomial multiplication and reduction at a time. To achieve the same throughput, one sample per cycle, a bit-level pipeline method is required, which results in a very short critical path at the cost of significantly more area and power. As we target the IoT application domain which does not require a very high throughput, our proposed design is a preferred choice. We support long bit-width products ($>$ 8-bits) by connecting multiple basic multiplication units together. This will be elaborated on in Section 4.2.4.3.

Square. The square operation is the other basic unit in our GF arithmetic unit. Although it is a specialized multiplication, we still implement it as a separate unit for two reasons. Firstly, square is much simpler than multiplication in Galois field. Mathematically, the full product of a square only spreads the input and inserts zeros in the odd positions as

Table 4.3: Comparison between Multiplication and Square

28nm	GF mult	GF square
m=5,6,7,8 ; arbitrary polynomial		
# of cells	263	73
Area (um^2)	199.59	63.48
Critical Path (ns)	0.4	0.2
Configuration of GF Arithmetic Unit		
# of primitive arithmetic units	16	28

shown in Fig. 4.5(c). Thus, a square operation only needs a polynomial reduction module. The hardware comparison between multiplication and square is illustrated in Table 4.3. Secondly, square is a heavily utilized operation to realize the complicated multiplicative inverse instruction. We will elaborate on this in Section 4.2.4.3.

4.2.4.2 Centralized Dedicated Configuration Register

We employ a dedicated centralized configuration register to support various bit-widths and arbitrary polynomials. It is set via a special configuration instruction when a Galois field is decided. Since it is shared among all arithmetic units, the overhead of supporting variable bit-widths and arbitrary polynomials are amortized.

The centralized configuration register is also used in data-gating almost half of the combinational logic when the 32-bit partial product instruction is executed, resulting in a 33% power reduction.

4.2.4.3 Complicated Instructions

Complicated instructions such as multiplicative inverse, 32-bit partial product, and SIMD instructions are realized by different connections among the primitive units in the interconnect fabric shown in Figure 4.4. Computation units are reused among different instructions, reducing area overhead.

Multiplicative Inverse. In Galois field, multiplicative inverse can be computed by $\alpha^{-1} = \alpha^{(2^m-2)}$, where α is an element in GF(2^8). This can lead to a large power depending on m. To reduce the number of power computations involved in this process, the Itoh-Tsujii algorithm (ITA) is employed [98]. The multiplicative inverse instruction is implemented by connecting a series of multiplications and squares. Four multiplications and seven squares are concatenated to realize one 8-bit multiplicative inverse. Fig. 4.6 presents an example for multiplicative inverse in GF(2^4). Multiplicative inverse over GF(2^3) is realized by muxing out the A^6 power.

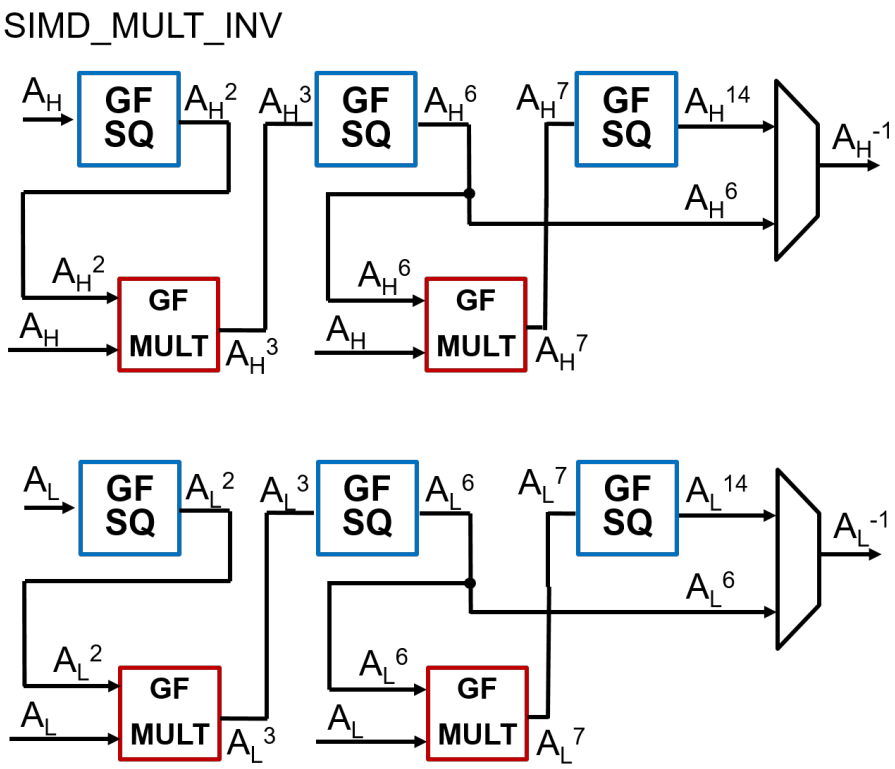


Figure 4.6: Multiplicative Inverse. Interconnecting primitive units to perform the multiplicative inverse.

We also considered other systolic methods [99, 100, 101], based on Euclid Algorithms (EA), which implement multiplicative inverse at a higher throughput. The hardware resource comparison between EA [100] and our method is illustrated in Table. 4.4. We implemented the ITA method for two reasons: first, we are not targeting a high throughput application domain; second, the ITA method can use existing multiplication/square hardware, requiring no extra area.

Single Cycle 32-bit Partial Product. A single cycle 32-bit partial product utilizes all 16 of the 8-bit GF multiplication units and a simple GF(2) addition (exclusive-OR). An example of 16-bit partial product is illustrated in Fig. 4.7. Very long bit width (¿100bit) GF multiplication benefits from this single cycle 32-bit partial product by iteratively using this product and performing a reduction step on the CPU. An example of efficiently supporting GF(2^{233}) multiplication is detailed in Section 4.3.3.4.

SIMD Instruction. We support SIMD instructions for multiplicative inverse (Figure 4.6), multiplication and square (Figure 4.8). Since there are 28 square units, the even-power instructions can be carried out in SIMD fashion with a few muxes, as shown in Figure 4.8 (top). We implement a four-lane 8-bit SIMD datapath for two reasons. First,

Table 4.4: Resource Comparison for Different Multiplicative Inverse Methods

	Systolic Euclidean Algorithm (pipelined)	This work ITA
Comb.	$m(6m+3)XOR+$ $m(6m+7)AND+$ $m(6m+5)MUX$	$(15m^2-11m)AND+$ $(15m^2-13m+4)XOR$
FF	$m(6m+4)$	No need
Total area* **	$57m^2$	$48.75m^2$
Configuration Datapath (shared by multiple ALUs)		
FF	m	$m(m-1)$
* AND : MUX : XOR : FF = 1 : 2.25 : 2.25 : 4 in a 28nm technology ** only m^2 item is considered, which overestimate the area of our work		

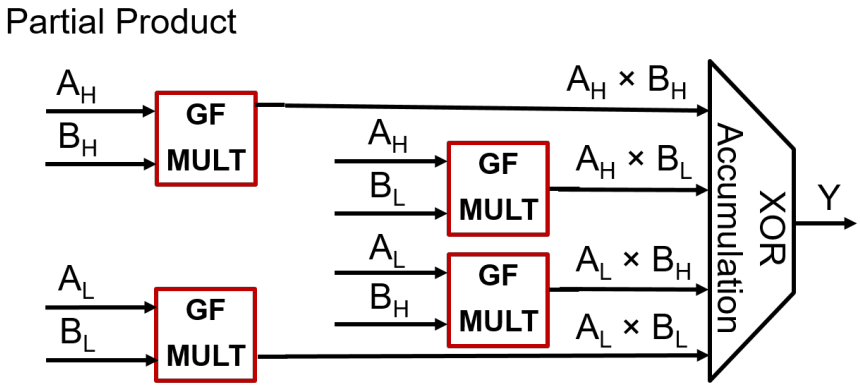


Figure 4.7: 16-bit Partial Product Example.The 32-bit case follows an analogous interconnection of primitive units.

there is limited parallelism in the applications, which will be discussed in Section 4.3.1. Second, we want to match the number of 8-bit GF multiplier involved in a 32-bit partial product and a SIMD multiplicative inverse. Both a 32-bit partial product and a four-way 8-bit SIMD multiplicative inverse can be realized with 16 8-bit full multipliers. Hence, a majority of the computation resources are reused by different instructions.

Data Gating of Idle GF Arithmetic Unit. The Galois field arithmetic unit is not always active. The GF instructions are interleaved with other instructions such as integer arithmetic and control instructions. Therefore, data gating is applied on the Galois field arithmetic unit by feeding zeros as default inputs, so the updating on the pipeline register will not incur switching activities in the GF arithmetic unit, resulting in a 77% dynamic power savings.

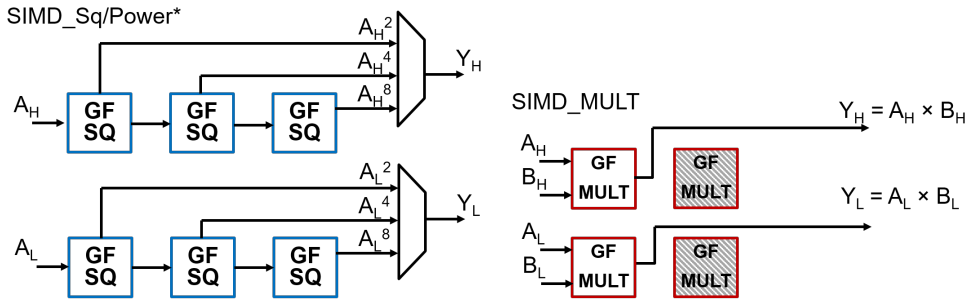


Figure 4.8: 2-way SIMD GF Multiplication and Square/Power Example. The 4-way case follows an analogous interconnection of primitive units.

4.3 Evaluation

4.3.1 Application Kernels

Table 4.5 illustrates the kernels we are evaluating for both cryptography and coding applications– here coding refers to the decoding process, while encoding is also feasible with the proposed architecture. We select the most common algorithms that works for all different types of RS/BCH codes. Even without any modification, the coding algorithms and AES reveal some degrees of parallelism [86]. We employ a four way SIMD to take advantage of this.

4.3.2 Kernel Mapping

Kernel mapping is performed by converting each step in ECC_r, AES, and ECC_l to a series of GF arithmetic operations and datapath control instructions.

The rest of this subsection is an example for syndrome calculation in the RS/BCH decoder datapath. This is the first and the only kernel that cannot be avoided in the decoding datapath. Because the decision whether to perform the rest of the decoding datapath relies on the value of syndromes. If syndromes are all zero, this indicates that no error occurred and the decoding will terminate.

Table 4.5: ECC$_r$, AES, and ECC$_l$ Kernel Algorithms with Levels of Parallelism.

Application	Kernel	Function Description	Parallelism Description
RS/BCH Decoder (n, k, t) $n = 2^{m-1}$ $GF(2^m)$	Syndrome Computation	Reflect the errors of the received message	Explicit vectorizable with 2t independent syndromes. Each syndrome computation is a n-iterative algorithm.
	Berlekamp-Massey Algorithm	Find the coefficients of error locator polynomial (ELP) $\Lambda(x) = 1 + \Lambda_1 x + \cdots \Lambda_t x^t$	Small and implicit parallelism. 2t' iterative algorithm to solve coefficients of t-degree polynomial ELP
	Chien Search	Find the root of ELP, which indicates the exact positions of errors	Explicit vectorizable with 2m independent elements evaluating in t-degree ELP
	Forney Algorithm	Calculate the error values at the error locations	Vectorizable with independent errors
AES	AddRoundKey	Combine the state with subkeys (in this round)	Vectorizable with 16 independent state bytes
	Sbox	Multiplicative inverse of each state bytes	
	Shift Row	Scramble along the row	Nonvectorizable data movement
	Mix Column	Scramble along the column	Vectorizable with 4 4-by-4 matrix vector operation
	Key Expansion	Generate the next subkey	Vectorizable with 4 (a row). Sbox the first row, then xor with all other rows
ECC$_l$	$GF(2^{233})$ Multiplication	Fundamental operations of point addition/point doubling over binary $GF(2^{233})$ $x^{233} + x^{74} + 1$	Utilize the single cycle 32-bit partial product Benefit from sparse polynomial reduction
	$GF(2^{233})$ Square		

Table 4.6: Syndrome Computation Comparison between Embedded GPP and Our Architecture

General Purpose Processor	This work
For j = 1,2,...,N	
$sumIdx = BIN2Idx[sum]$ $sumIdx = (sumIdx + i)\%fieldsize$ $sum = Idx2BIN[sumIdx] \bigoplus R_j$	$sum = sum \bigotimes \alpha^i$ $sum = sum \bigoplus R_j$

As illustrated in Table 4.5, there are $2t$ independent syndromes to compute for t-bit error correction. Syndrome computation is parallelized in a straightforward manner. Each syndrome S_i is typically computed using Horner's rule via the recursive calculation in the form of $S_{i,j} = S_{i,j-1}\alpha^i + R_{n-j}$ for $j = 1, 2, ..., n$ where $S_{i,j}$ is the computed syndrome after $j-$rounds of recursion, α is a primitive element in Galois field, and R_{n-j} is the received symbol in the codeword. $S_i = S_{i,n}$ holds when n is the number of symbols in a codeword. The inner loop for syndrome calculation requires one GF multiplication and one GF addition. On a general purpose processor, the multiplication is typically optimized by performing the calculation in the log domain [102] that requires table lookup operations, as shown in left side of Table 4.6. With our architecture, the GF operations will directly call the corresponding instruction, as shown in the right side of Table 4.6.

The left column of Table 4.6 indicates that each inner loop of syndrome calculation involves one integer, one modulo, one bitwise exclusive-OR, and two multi-cycle table lookup operations. In contrast, in our architecture, the inner loop involves just two single cycle GF operations.

Because the inner loop computation on our architecture avoids table lookup operations, we also achieve reduced data memory footprint. Furthermore, because we have less variables required for each inner loop, our register files have less pressure. Finally, the $2t$ syndrome computations can be vectorized in a straightforward manner, as discussed in the beginning of this section. The cumulative effect of these improvements results in an over $20\times$ speedup.

4.3.3 Performance

4.3.3.1 Methodology

The baseline for our evaluation is obtained by running the kernels from Table 4.5 in Keil uVision 5, an ARM integrated development environment. It includes both a compiler and cycle accurate simulator. The target platform is a Cortex M0+, a two-stage in-order processor [59]. We are able observe the assembly code in the simulator. To obtain performance

results for our architecture, we take the assembly code that performs a Galois field operation, for example the left column in Table 4.6, and replace it by our GF instructions. The control code and other non-GF arithmetic operations are kept the same. The reduced register pressure allows us to hold more data elements in the unoccupied registers. The cycles to compute each GF instruction using our architecture are deterministic and added to the simulated runtime.

4.3.3.2 BCH/RS Decoder

We implemented a decoder consisting of syndrome calculation, Berlekamp-Massey-algorithm (BMA), Chien search and Forney's algorithm. The baseline implementation on the M0+ is optimized by conducting GF multiplication/ multiplicative inverse in the log domain [102].

On our architecture, we do not need to operate in the log domain because all GF operations are directly performed on the GF arithmetic unit. A binary BCH (31,11,5) decoder is realized by combining the first three kernels and setting $n = 31$, $k = 11$, and $t = 5$ in the algorithm. An RS (255,239,8) decoder is realized by utilizing all four kernels (syndrome calculation, BMA, Chien search, Forney's algorithm) and setting the parameters to $n = 255$, $k = 239$, and $t = 8$. The speedup is shown in Fig. 4.9.

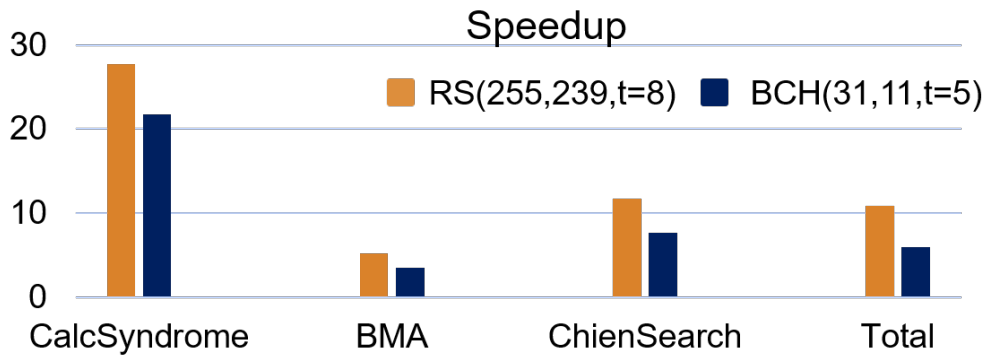


Figure 4.9: ECC$_r$ Decoder Speedup over M0+. Specifically comparing against RS on GF(256) and BCH on GF(32).

Computation dominant kernels, such as syndrome calculation, are very efficient on the proposed architecture. RS(255, 239, 8) has a better speedup compared to BCH(31,11,5) because RS($t = 8$) has 16 independent syndromes, a perfect match for our 4-way SIMD, while BCH($t = 5$), with 10 independent syndromes, looses two lanes in the last round.

BMA experiences the least speedup because it is an iterative algorithm with limited parallelism in the inner loop. In this evaluation, we only exploit the most straightforward parallelism in computing several intermediate t-degree polynomial coefficients. As dis-

cussed in Table 4.5, Chien search has many independent elements to evaluate. But for each element, the computation complexity is proportional to the degree t. Comparing it to the complexity $n(> t)$ of syndrome computation, the control overhead of Chien search is larger.

Forney's algorithm for RS codes has over a 10× speedup. Forney's algorithm evaluates each error location using the GF multiplicative inverse and multiplication operations, which all map to single cycle operations on our architecture. We are able to calculate four independent errors in parallel.

The overall speedup for RS is more than 10×, which is better than the binary BCH decoder speedup. Forney's algorithm is not required in the binary BCH decoder because the error is binary. However, the binary BCH exposes another level of parallelism—inter-codeword parallelism. This happens because it creates numerous shorter codewords in one packet. Potentially, the inter-codeword parallelism embedded in binary BCH could lead to a higher performance gain if we had attempted to optimize for it.

4.3.3.3 Symmetric Cryptography: AES

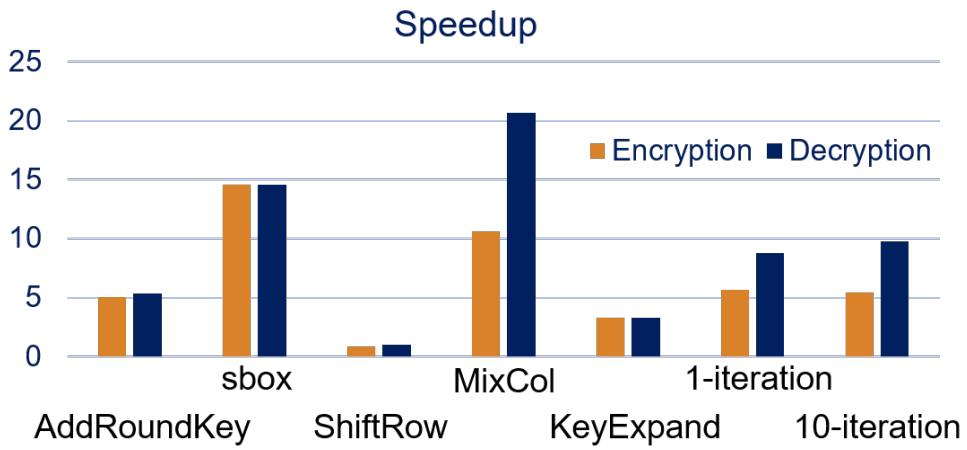


Figure 4.10: AES Speedup over M0+

Several open source benchmarks for AES [103] [104] [105] are compared. The implementation in [103] achieves the best performance on the baseline ARM M0+ platform.

Across all the kernels, S-Box and MixCol/invMixCol show the best speedup. Traditionally, S-Box is implemented as a table lookup operation while in our architecture it is realized directly with the multiplicative inverse operation. Similarly, MixCol/invMixCol in our architecture are also realized directly by performing inner products with Galois field

arithmetic. The MixCol and invMixCol can use the same code but with different coefficients.

We profiled these open source benchmarks, and observed that MixCol/invMixCol is the hotspot. The selected baseline benchmark on the ARM M0+ obtains the best performance mostly because of good MixCol/invMixCol optimization. MixCol shows good optimization because the coefficients for MixCol are $0x02$, $0x03$, $0x01$, and $0x01$. GF multiplication with these values can be optimized into a few exclusive-ORs and shifts. While for invMixCol, the coefficients are $0x11$, $0x13$, $0x09$, and $0x14$. As a result, the data dependent optimizations on the ARM M0+ are not as efficient as MixCol. Our system obtains a more than a $10\times$ speedup on MixCol, and, because our system is agnostic to the values of the coefficients, we achieve an even higher $20\times$ speedup on invMixCol kernel. Overall, gains of more than $5\times$ for encryption and more than $10\times$ for decryption are achieved, mostly from improvements in MixCol/invMixCol and S-Box.

4.3.3.4 Asymmetric Cryptography

Multiplication/Square on GF(2^{233}). Long bit width(e.g. 233-bit) GF multiplication and square are two essential operations for ECC_l. The literature states that GF multiplication typically accounts for most of the execution time [106], approximately 80% [107, 108]. We utilize the same ITA method described in Section 4.2.4.3 to realize the multiplicative inverse operation. We hand-code these operations on GF(2^{233}) with a NIST Koblitz curve $x^{233} + x^{74} + 1$ [92].

The multiplication on GF(2^{233}) is achieved by performing two steps. First, the full partial product of two 233-bit inputs (8 words with 32bits/word) is computed and stored back to memory (16 words) after this procedure.

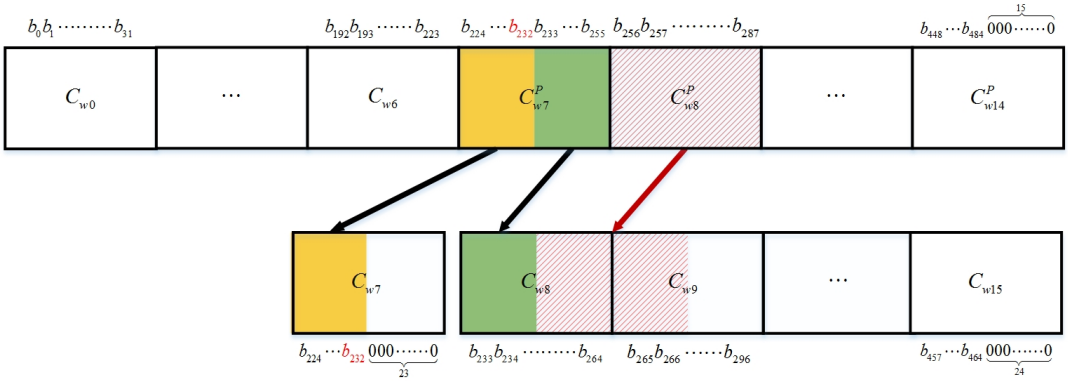


Figure 4.11: Rearrange the partial product into the reduction vector and the remaining vector.

Table 4.7: Operation Cycle Count Breakdown of ECC_l Multiplication and Square on GF(2^{233}) over NIST Koblitz Curve $x^{233} + x^{74} + 1$

Operation	LD	ST	GF 32b Partial Porduct	ALUs	Total cycle
233-bit multiplication					
Full Product	72	71	64	112	462
Rearrange	8			29	45
Polynomial Reduction	8	8		60	92
Total Operation	88	79	64	201	
Total Cycle	334		64	201	599
233-bit square					
High Full Product & Rearrange	5	2	5	30	49
Low Full Product & Reduction	5	8	3	58	87
Total Operation	10	10	8	88	
Total Cycle	40		8	88	136
*ALUs includes bitwise ops such as AND,SHIFT,XOR. **GF addition is identical to BITXOR, sorted in ALUs. ***LD/ST has 2-cycle; other operations are all single cycle					

Table 4.8: ECC$_l$ GF Multiplication/Squaring on Different Platforms

	Erdem [110]		Clercq [109]	This work
Platform	ARM7TDMI		Cortex M0+	2-stage pipeline in-order proc. with GF unit
GF(*)	2^{228}	2^{256}	2^{233}	
Mult.	4359	5398	3672	599
Square	348	389	395	136

The second step is to perform the polynomial reduction on the partial product in the CPU. We rearrange the partial product into the reduction vector and the remaining vector. We place the lowest 233 bits (c_0-c_{232}) into the first 8 words of memory and pad it with 23 zeros. The remaining bits (c_{233}-c_{464}) are padded with 24 zeros to create the reduction vector in the last 8 words of memory. This rearrangement processing is illustrated in Figure 4.11. The polynomial reduction operates on one 32-bit word at a time with only a few shift and exclusive-OR operations. The cycle breakdown of a 233-bit multiplication for ECC$_l$ is illustrated in Table 4.7.

The square operation follows the same procedure as multiplication. We also leverage the simple structure of the square discussed in Section 4.2.4.1. Only 8 32-bit partial products are necessary to achieve a square partial product. Memory operations to store intermediate results are largely avoided. We interleave the full partial product operations and then rearrange results together for square. The cycle breakdown of a 233-bit square operation is presented in Table 4.7.

Table 4.8 illustrates the performance comparison to other works. We achieve a 6.1$\times$ speedup on multiplication and a 2.9$\times$ speedup on squaring for the same GF(2^{233}) field. Typical software optimizations involve pre-computed tables in memory, incurring a large storage overhead, which is undesirable for low power devices. For example, at least 4KB table storage is required in [109]. In our implementation, table storage is entirely avoided. We observe that on our platform, the memory operations also dominate. However, we take the most straightforward method to implement the algorithms without significant effort on software optimization. In fact, the methods illustrated in [109], which optimize to avoid memory accesses, will be able to be applied on top of our architecture. We provide architectural augmentation on the arithmetic level for Galois field operations, which is orthogonal to register scheduling optimizations or other pure software optimizations.

GF(2^{233}) Multiplication with Karatsuba Software optimization. Karatsuba algorithm [111] is a software optimization to compute the product of two large numbers. Suppose A, B are w-bit numbers, $C = A \times B$ is the target product. A_h, A_l, B_h, B_l are $w/2$-bit

Table 4.9: Operation Breakdown Comparison between direct product and Karatsuba optimization of a full multiplier of two 2^{233} numbers.

		MEM		GF mult	regular ALUs	Total
		LD	ST			
233-bit mult Op Count	Direct (Original)	72	71	64	112	–
	Karatsuba -opt	40	35	48	104	–
233-bit mult Cycle Count	Direct (Original)	286		64	112	462
	Karatsuba -opt	150		48	104	302

numbers. A_h, B_h are the high $w/2$-bit of A, B and A_l, B_l are the low $w/2$-bit of A, B, i.e., $A = A_h 2^{w/2} + A_l$ and $B = B_h 2^{w/2} + B_l$. In contrast to the direct product method, which requires four $w/2$-bit multiplications ($A_h B_h, A_h B_l, A_l B_h, A_l B_l$) and four $w/2$-bit additions, the Karatsuba algorithm employs three $w/2$-bit multiplications and six $w/2$-bit additions: $C = a2^w + e2^{w/2} + d$, where $a = A_h B_h, d = A_l B_l, e = (A_h + A_l)(B_h + B_l) - a - d$. The three multiplications compute a, d, and e respectively. The Karatsuba algorithms can be recursively called by further breaking the multiplication into $w/4, w/8, ...$, etc. Performance will benefit when the cost of one multiplication is greater than the cost of two additions. In this work, we employ a two-level Karatsuba algorithm to perform the product of two 2^{233} numbers. The operations breakdown of this two-level Karatsuba optimization is illustrated in Table 4.9. The same method shown in Table 4.7 for polynomial reduction is applied. Adding the 137 cycles of a polynomial reduction, the Karatsuba optimized $GF(2^{233})$ multiplication obtains a speedup of $1.4\times$ comparing to the direct method on our architecture, and a speedup of $8.4\times$ compared to the baseline implementation.

Point Addition/Point Doubling. We implemented point addition and point doubling on projective coordinates [90]. They require only GF multiplication, square and addition operations. However, translating points from affine coordinates, in which coordinates directly map to a point on an elliptic curve, to projective coordinates, requires the GF inverse operation. This translation is called once per scalar multiplication.

The cycle counts to perform point addition/point doubling and multiplicative inverse are given in Table 4.10. The comparison point, Clercq [109, 108] in Table 4.10, is implemented on an ARM M0+ platform. Our architecture with the direct product method achieves a $5.1\times$ improvement for point addition, and a $3.5\times$ improvement for GF(2^{233})

Table 4.10: Cycle Counts of ECC$_l$ Point Addition/Point Doubling

NIST Koblitz Curve-K233		Clercq [109, 108]	This Work direct product	This Work Karatsuba-opt
Level 1	Multiplication	3672	599	439
	Multiplication Precompute	675	–	–
	Addition	68	66	66
	Square	395	136	136
Level 2	Point Addition	34,426	6,742	5,302
	Point Doubling	Not reported	3,499	2,859
Level 2	Inverse	139,000	39,972	38,372

multiplicative inverse. Our architecture with the Karatsuba software optimization achieves a 6.5× improvement for point addition, and a 3.6× improvement for GF(2^{233}) multiplicative inverse.

Scalar Multiplication and ECC$_l$ Applications. Scalar multiplication on an elliptic curve is the operation of adding a point on the elliptic curve to itself repeatedly, i.e. $kP = P+P+P+...+P$ for a scalar k and a point $P = (P_x, P_y)$ on the elliptic curve. We employ the *double-and-add* method to implement scalar multiplication. For example, $k = 15$, kP is computed as $15P = 2[2(2P + P) + P] + P$, which requires 3 Point Additions and 3 Point Doublings.

The Elliptic Curve Diffie Hellman (ECDH) key exchange protocol [112] is illustrated in Figure 4.12. Alice has her own private key d_a (a scalar) and public key Q_a (a point on an elliptic curve). Meanwhile, Bob has his own private key d_b (a scalar) and public key Q_b (a point on an elliptic curve). A public key Q equals to dG, where d is the private key (the scalar) and G is a base point on the elliptic curve, which is known to both parties. Firstly, Alice and Bob will send their public key to each other. Once Alice received Bob's public key Q_b, she will compute the shared key by performing one scalar multiplication d_aQ_b. Similarly, Bob will perform the scalar multiplication d_bQ_a. The coordinates of d_aQ_b and d_bQ_a are identical because $d_aQ_b = d_ad_bG = d_bd_aG = d_bQ_a$, and this is the shared secret between Alice and Bob. Typically, the x-coordinate is used as the shared key for future data encryption/decryption with a symmetric cryptography method. For example, the x-coordinate can be used as a private key in AES. Alice and Bob's public keys—Q_a, Q_b—are transmitted as plaintext without any encryption, so the public keys can be acquired by anyone who is listening on the channel. However, without knowing Alice's and Bob's private key, the people in-the-middle need to solve a discrete logarithm problem to obtain the shared secret between Alice and Bob.

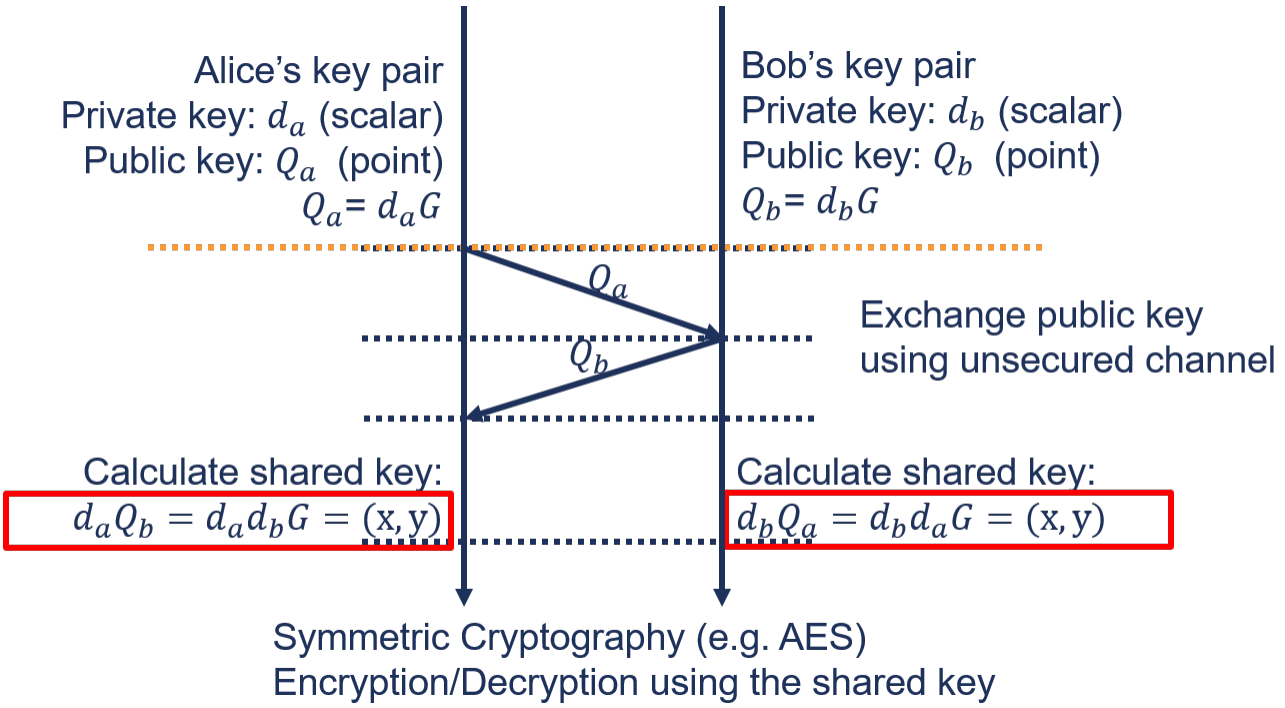


Figure 4.12: ECDH key exchange protocol is dominated by one scalar multiplications.

The control operations to execute this ECDH protocol are negligible compared to the workload to compute a scalar multiplication. On the standard $GF(2^{233})$ Koblitz curve and assuming a 112-bit level security (the highest bit of the scalar k is one and the remaining 112 bits consist of 56 zeros and 56 ones), the supporting functions, including two multiplicative inverses to perform coordinate projection, consume 157,442 cycles. The 56 Point Additions and 112 Point Doublings take 617,120 cycles to perform. Running at $100MHz$, our GF processor takes 7.75ms for one scalar multiplication and the ECDH key exchange protocol finishes within 8ms.

4.3.4 Power and Area

We implement the RTL-level design of our Galois field arithmetic unit and the two-stage in-order processor. The design is synthesized in a 28nm technology. Prime-Time/Power is used to estimate critical path delay and power consumption, respectively. The instruction sequences contain both program control instructions (from the ARM compiler) and hand-coded GF instructions. Data inputs are random generated. The switching activity is dumped from the testbench and fed into Prime-Time/Power to generate a power/time estimation using the synthesized netlist.

Table 4.11: Power and Area of the Galois Field Arithmetic Unit

m=5,6,7,8; arbitrary polynomial Configuration of GF Arithmetic Unit			
28nm	GF mult	GF sq	Inst. Control
# of primitive unit	16	28	–
Single primitive unit area (um^2)	199.59	63.48	–
Total Area (um^2)	3193	1777	1005
	5760		
Critical path (ns)	2.91ns @ GF multiplicative_inverse		

Table 4.12: Proposed GF Processor Characteristics

28nm @0.9V 100MHz		Gate count	Area (um^2)	Power (uW)
2-stage Processor Shell	Comb.	3482	2258	
	Register File	694	2254	
	Total	4176	4512	279
GF Arithmetic Unit		7494	5760	152
Design total		11670	10272	431

4.3.4.1 Galois Field Arithmetic Unit

The 16 GF multiplier has $3193um^2$ area, 28 GF square has $1777um^2$ area, the design total area is $5760um^2$. The critical path is 2.91ns, which happens at GF multiplicative inverse. The results are illustrated in Table 4.11.

The results indicate that our Galois field arithmetic unit is compact, less than $6000\mu m^2$, and fast enough, with a critical path of 3ns, to support applications in the IoT domain. We believe this Galois field arithmetic unit is a basic building block that can be integrated into many embedded processors as a hardware accelerator.

4.3.4.2 The Galois Field Processor

As an example, we integrate the GF arithmetic unit into a two stage in-order general purpose processor with a 16-entry 32-bit register file. The total area is $10{,}272\mu m^2$ and consumes 431 μW at nominal voltage 0.9V while running AES at $100MHz$. The GF processor can run at a maximum clock upto $300MHz$, which is more than enough for the low power, low bandwidth IoT application domain that we considered in this work. Thus a deeper pipeline, which can provide more throughput with a higher frequency and consumes more energy per instruction, wasn't considered as our control logic architecture.

We conduct SPICE simulation on this technology and apply voltage scaling optimizations to our design. The results show that the design can be voltage scaled down to 0.7V for a target frequency of 100MHz. When scaling, we leave a more than 50% time margin to account for increased variability at lower voltages. This time margin is more than that of previously fabricated designs in 28nm [113] that employ voltage scaling. At 0.7V, the GF arithmetic unit consumes 75μW, and the processor consumes 231μW in total. Voltage scaling will improve our energy efficiency by 1.86×.

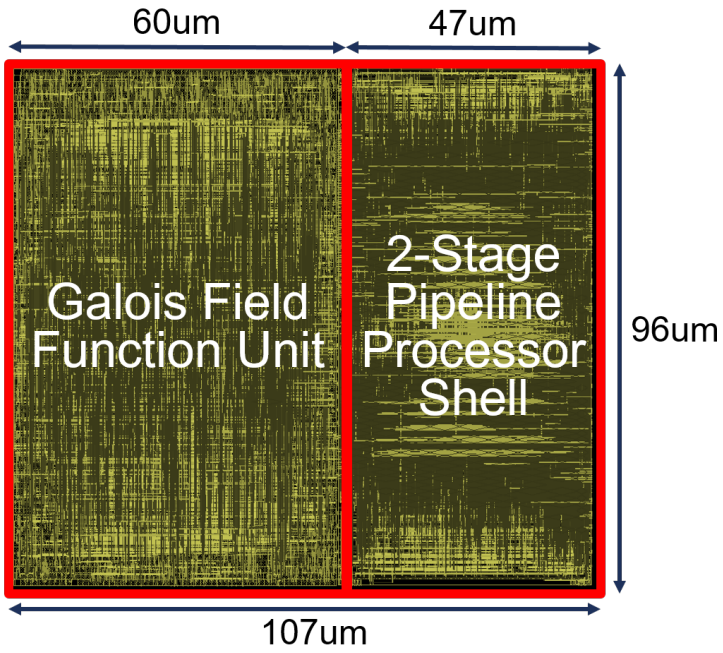


Figure 4.13: Layout. The area is 10,272um^2.

Scaling the GF processor. Our preferred design has 16 GF MULT units and 28 GF SQ units. The four-way SIMD multiplicative inverse and 32-bit partial product both require 16 GF MULT units for single cycle operations. 28 GF SQ units are needed to support a single cycle SIMD multiplicative inverse. More mult units would benefit elliptic curve cryptography applications, but result in more idle hardware for error correction codes due to limited parallelism. Thus this configuration provides a reasonable area/efficiency trade-off points considering across the applications. Different configurations of primitive units as well as the general purpose control logic units can be re-evaluated. For example, the IoT device may operate as a controller node in local networks and may need to authenticate with several end nodes, then the elliptic curve cryptography applications are more likely to be performed within tighter latency. In this application scenario, increasing RF entries and GF primitive units will improve the performance of ECC$_l$ applications. Table 4.13 illus-

Table 4.13: ECC$_l$ applications performance with different system configurations.

	Current	Double RF	Double alu/RF
	4-way 32-bit Datapath		2x 4-way 32-bit
	# of primitive units/RF entry		
GF mult	16	16	32
GF sq	28	28	56
RF	16' 32-bit	32' 32-bit	32' 32-bit
Total Area (μm^2)	9586	11,838 23.5% more	16,830 76% more
233-bit Mult	439 cyc 8.4×	300 cyc 12.2×	194 cyc 20×
233-bit Square	136 cyc 2.9×	128 cyc 3.1×	96 cyc 4.1×
Point Addition	5302 cyc 6.5×	4019 cyc 8.6×	2909 cyc 11.8×
Point Doubling	2,895 cyc	2,263 cyc	1,667 cyc
233-bit MultInv	38,372 cyc 3.6×	35,126 cyc 4×	26,642 cyc 5.2×
112-bit security Scalar Mult.	775k cyc	639k cyc	484k cyc

trates the performance comparison among different system configurations. The doubling RF entries configuration speedup the scalar multiplication by 20% with 23% area overhead while the configuration doubling both ALU and RF entries speedup the scalar multiplication by 60% with 75% area overhead. The proposed GF processor can be scaled to other system configurations if the target application areas migrate.

4.3.5 Comparison to ASICs

Comparing to AES. Compared to the smallest AES ASIC from Intel [114], our Galois field arithmetic unit, which supports both encryption and decryption, is smaller than the total area of the encryption and decryption datapath combined, as illustrated in Table 4.14. With 63.5% additional area in total, our processor can support not only the different modes of AES, but also various error correction codes and efficient arithmetic support for asymmetric cryptography–ECC$_l$.

On the other hand, programmable solutions are known to consume more power than their ASIC counterpart. Our design consumes 6× more energy per bit compared to the most energy efficient compact AES ASIC design [115]. For systems that are extremely power constrained (i.e., 100s of μW), the ASIC remains a more suitable choice. How-

Table 4.14: Area Comparison with Smallest AES ASIC

Area (um^2) Scaled to 28nm	Intel [114]	Encryption	Decryption	Total
		2800	3482	6282
	This work	GF arithmetic unit		Total
		5760		10272

Table 4.15: Comparison with the Most Energy Efficiency Compact AES ASIC

Scaled to 28nm 0.9V 100MHz	Power (uW)	Throughput (Mbps)	Energy Efficiency (pJ/b)
Zhang [115]	236	38	6.21
This work	431	12.2	35.5

ever, if the system needs flexibility, than our design remains a better choice than using the CPU to support that flexibility. To achieve flexibility with an ASIC would require placing multiple ASIC's on the die, consuming area and increasing cost. Moreover, our solution allows flexible coding and cryptography algorithms to be updated. So for flexible designs concerned with area constraints, our design is preferred.

We do not present any other complete energy efficiency and area comparison at the application level. Other coding and asymmetric cryptography ASIC solutions [77, 78, 79, 116] include implementations with very different features that are either not reported or ambiguous. However, this does not affect the conclusion that a multiple ASIC solution will consume more area than our solution, given the smallest AES [114] is over 60% of our total area.

Comparison to GF Multiplier ASIC. The work in [117] presents a single cycle 64-bit Galois field multiplier for on-die acceleration of public key encryption for a high throughput processor. After scaling it down to 28nm, this GF multiplier accelerator consumes 1.25 mW at 0.9V and 100 MHz, while our GF processor including both arithmetic unit and control logic consumes 0.43 mW under the same condition. The area of our GF processor is 77% of this 64-bit GF multiplier accelerator. In addition, our solution can be programmed to support smaller bit-width operations, which are critical for flexible coding support. Our proposed processor consumes 4$\times$ energy per 64b multiplication operation partially due to the register file and control overhead and to the extra cycles for accumulation.

In summary, comparing to ASIC solutions, we provide programmability to support various algorithms in different applications. The unified underlying datapath allows us to address the flexibility within a very small area. The efficient arithmetic support maintains the cost of this flexibility within an affordable region for an IoT device.

4.4 Related Works

In previous works, flexible designs were only employed within a particular application (RS, AES, or ECC$_l$), and not optimized to support functionality across the entire GF application space. Previous work [96][97] focused on binary Galois field arithmetic optimizations for very long bit widths, targeting for only the ECC$_l$ domain. Moreover, they focus on achieving very high throughput, which is different from the IoT space that this work focuses on. Previous works, typified by [118], propose configurable and high throughput solutions for symmetric cryptographic algorithms with efficient table lookup support. [119] addresses both AES and ECC$_l$ in the same processor. In their work, AES and ECC$_l$ have an individual computation datapath, while the control and memory are shared among two cryptographic methods. This is different from our implementation, where AES and ECC$_l$ utilize the same computation datapath by sharing small bit-width SIMD operations and long bit-width operations. Compared to the work of [120, 121, 122], which supports RS coding with various parameters, we can perform not only RS codes but also other coding schemes in Galois fields. The work in [123] presents optimization for ECC$_l$ on both binary and prime domain, but only on large bitwidths. The low power IoT devices typically prefer binary Galois Field due to its neat implementation, thus we don't consider to support a prime Galois Field. However, we are not just designing in the large GF for ECC$_l$ applications. We design for a broader application domain that includes both small and large bitwidth for both ECC$_l$, ECC$_r$, and AES applications.

4.5 Conclusion

In this work, we presented a light-weight Galois field processor, providing efficient support for both flexible information coding and secure connectivity on IoT devices. We described the design challenges and proposed a flexible bitwidth Galois field arithmetic unit composed of two types of primitive units. Complicated instructions, including multiplicative inverse and long bit-width partial product were supported by selectively connecting these primitive units. Configurable SIMD instructions were implemented to exploit the parallelism in coding and cryptography. The arithmetic unit was integrated into a two-stage in-order processor. Several coding and symmetric/asymmetric cryptography kernels were mapped onto our architecture, exhibiting a $5-20\times$ speedup across kernels. We demonstrated the effectiveness of our GF processor by synthesizing it in a 28nm technology, where it consumed just $431\mu W$ at 0.9V and 100MHz, while only requiring a place/route area of 0.0103 mm^2.

Finally, we demonstrated the feasibility of a unified architecture to enable coding flexibility and secure wireless communication in the low power IoT domain. The energy efficiency of IoT wireless communication can be enhanced by enabling dynamic adaptation to the error coding schemes in various channel conditions on the proposed architecture. Meanwhile, secure communication was realized via efficient support for both asymmetric and symmetric cryptography on the same hardware platform.

CHAPTER 5

Conclusions

As the number of IoT devices increases rapidly and many billions of devices communicate with each other and connect to the Internet, wireless communications will play a critical part. This dissertation investigated both the system design and architecture design for flexible, energy efficient and secure wireless communications solutions for two categories of IoT devices: ultra-small millimeter-scale IoT nodes and standard compliant IoT devices.

For the ultra-small millimeter-scale IoT nodes, on which none of the standard communications can be supported, conventional communication technology cannot be applied directly. Our solution was an energy-autonomous, self-contained wireless communications system design, presented in Chapter 2, to optimally utilize the scarce energy/power resources. In Chapter 2, the unique millimeter-scale communication system design considerations were discussed at first. Then a new synchronization protocol was proposed by exploiting the system asymmetry between sensor nodes and the gate-way, and a new modulation scheme was specialized for the ultra-small IoT nodes. The detailed mathematical models of modulation, coding, synchronization, energy consumption and data rate of the proposed system were presented. The dynamic link adaptation problem for various objective functions were formulated mathematically. The simulations were conducted to quantify the optimum results, and the impact of different system configurations were analyzed to guide system designers toward the energy optimized communication system for ultra-small IoT sensor nodes. In a system where all components including the battery, energy-harvesting cells, the RF antenna, reservoir capacitors, transceiver circuits, etc, are integrated within a millimeter-scale form factor, we show an energy optimized wireless communication system design can obtain orders of magnitude improvements in wireless communication energy efficiency.

The focus of the remaining chapters of this dissertation was on standard compliant wireless communications in the IoT application domain. Two architectural designs were presented: the IoT SDR baseband processor and the Galois field processor.

The IoT SDR architecture was proposed as a programmable and low power solution for baseband processing in Chapter 3. The fundamental algorithms in communications systems on IoT devices were analyzed and then used to define a microarchitecture design that supports many IoT standards and custom nonstandard communications. The solution that resulted was a custom SIMD execution machine coupled with a scalar unit. Several architectural optimizations were introduced including efficient streaming computation, flexible bit-width, optimized vector reduction operations as well as low power techniques such as voltage scaling and clock gating. A comprehensive evaluation on power and area was conducted. The single IoT SDR datapath is placed and routed to fit an area of $0.074mm^2$ with a sub-mW power consumption in a 28nm technology. Two representative upper bound systems, 802.15.4-OQPSK and BLE were mapped onto the IoT SDR architecture. The peak power of the proposed full baseband system is below $2mW$, which is only 8% of the total system power, demonstrating the value of baseband SDR in the low power IoT domain.

The light-weight Galois field processor in Chapter 4 was presented to provide efficient support for both flexible information coding and secure connectivity on IoT devices. Several error correction codes and cryptography kernels were analyzed to identify the shared fundamental operations and the design challenges. A flexible bitwidth GF arithmetic unit composed of two types of primitive units—multiplication and square—were proposed. The single cycle complex instructions that include multiplicative inverses and 32-bit GF product are implemented by different connections of primitive units. These configurable four-way 8-bit SIMD instructions were employed to exploit the parallelism in coding and cryptography datapaths. The extremely long bit-width ($> 100bit$) multiplication was efficiently handled by iteratively utilizing a single-cycle 32-bit multiplier. The GF arithmetic unit is integrated into a two-stage in-order processor. This GF processor exhibits a 5-$20\times$ speedup across a range of error correction codes and symmetric/asymmetric cryptography applications. This GF processor consumes $431uW$ at $0.9V$, $100MHz$ while running AES at $12.2Mbps$ in a 28nm process within an area of $0.01mm^2$. Thus, the feasibility of a unified processor architecture to enable coding flexibility and secure communication in low power IoT wireless was demonstrated. By adapting the coding parameters for various noise/interference conditions, the energy efficiency of IoT wireless communications can be enhanced. Meanwhile, both symmetric and asymmetric cryptography processing can be efficiently supported on the same underline hardware.

A wireless system architecture combining both an SDR baseband processor and a light weight Galois field processor is able to: 1) address many standard and nonstandard communications; 2) capture standard evolution by software updating; and 3) obtain better energy efficiency by dynamically adapting system configurations, such as data rate, modulation

schemes, and coding parameters, to different operating scenarios. At the same time, security aspect of IoT communications can be realized by efficient implementation of both symmetric and asymmetric cryptography applications on the same GF processor.

In conclusion, designing flexible, energy efficient and secure wireless communications solutions are essential in the area of IoT applications.

BIBLIOGRAPHY

[1] Andrew C. Russell, N. C., "NXP and the Internet of Things (IoT)," .

[2] "IEEE Standard 802.15.4 Part 15.4: Low-Rate Wireless Personal Area Networks," 16 June 2011.

[3] "IEEE Standard 802.15.6 Part 15.6: Wireless Body Area Networks," 29 February 2012.

[4] "Bluetooth SIG. Bluetooth Specification Version 4.0," *The Bluetooth Special Interest Group*, 30 June 2010.

[5] "Bluetooth 5 Core Specification," .

[6] "802.15.4q-2016 - IEEE Standard for Low-Rate Wireless Networks –Amendment 2: Ultra-Low Power Physical Layer," .

[7] "LoRA Alliance," .

[8] Lee, Y., Kim, G., Bang, S., Kim, Y., Lee, I., Dutta, P., Sylvester, D., and Blaauw, D., "A modular 1mm 3 die-stacked sensing platform with optical communication and multi-modal energy harvesting," *2012 IEEE International Solid-State Circuits Conference*, IEEE, 2012, pp. 402–404.

[9] Kim, G., Lee, Y., Foo, Z., Pannuto, P., Kuo, Y.-S., Kempke, B., Ghaed, M. H., Bang, S., Lee, I., Kim, Y., Jeong, S., Dutta, P., Sylvester, D., and Blaauw, D., "A millimeter-scale wireless imaging system with continuous motion detection and energy harvesting," *2014 Symposium on VLSI Circuits Digest of Technical Papers*, June 2014, pp. 1–2.

[10] Warneke, B., Last, M., Liebowitz, B., and Pister, K. S. J., "Smart Dust: communicating with a cubic-millimeter computer," *Computer*, Vol. 34, No. 1, Jan 2001, pp. 44–51.

[11] Schwiebert, L., Gupta, S. K., and Weinmann, J., "Research Challenges in Wireless Networks of Biomedical Sensors," *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, MobiCom '01, ACM, New York, NY, USA, 2001, pp. 151–165.

[12] Chow, E. Y., Chakraborty, S., Chappell, W. J., and Irazoqui, P. P., "Mixed-signal integrated circuits for self-contained sub-cubic millimeter biomedical implants," *2010 IEEE International Solid-State Circuits Conference - (ISSCC)*, Feb 2010, pp. 236–237.

[13] Mohamed, N. and Jawhar, I., "A Fault Tolerant Wired/Wireless Sensor Network Architecture for Monitoring Pipeline Infrastructures," *Sensor Technologies and Applications, 2008. SENSORCOMM '08. Second International Conference on*, Aug 2008, pp. 179–184.

[14] Elvin, N. G., Lajnef, N., and Elvin, A. A., "Feasibility of structural monitoring with vibration powered sensors," *Smart Materials and Structures*, Vol. 15, No. 4, 2006, pp. 977–986.

[15] Tachwali, Y., Refai, H., and Fagan, J. E., "Minimizing HVAC Energy Consumption Using a Wireless Sensor Network," *Industrial Electronics Society, 2007. IECON 2007. 33rd Annual Conference of the IEEE*, Nov 2007, pp. 439–444.

[16] Demirkol, I., Ersoy, C., and Onur, E., "Wake-up receivers for wireless sensor networks: benefits and challenges," *IEEE Wireless Communications*, Vol. 16, No. 4, Aug 2009, pp. 88–96.

[17] Ansari, J., Pankin, D., and Mahonen, P., "Radio-Triggered Wake-ups with Addressing Capabilities for extremely low power sensor network applications," *2008 IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications*, Sept 2008, pp. 1–5.

[18] Chen, G., Ghaed, H., u. Haque, R., Wieckowski, M., Kim, Y., Kim, G., Fick, D., Kim, D., Seok, M., Wise, K., Blaauw, D., and Sylvester, D., "A cubic-millimeter energy-autonomous wireless intraocular pressure monitor," *2011 IEEE International Solid-State Circuits Conference*, Feb 2011, pp. 310–312.

[19] Jung, W., Oh, S., Bang, S., Lee, Y., Foo, Z., Kim, G., Zhang, Y., Sylvester, D., and Blaauw, D., "An Ultra-Low Power Fully Integrated Energy Harvester Based on Self-Oscillating Switched-Capacitor Voltage Doubler," *IEEE Journal of Solid-State Circuits*, Vol. 49, No. 12, Dec 2014, pp. 2800–2811.

[20] Best, S. R. and Yaghjian, A. D., "The lower bounds on Q for lossy electric and magnetic dipole antennas," *IEEE Antennas and Wireless Propagation Letters*, Vol. 3, No. 1, Dec 2004, pp. 314–316.

[21] Balanis, C. A., *Antenna theory: analysis and design*, John Wiley & Sons, 2016.

[22] "Propagation data and prediction methods for the planning of indoor radiocommunication systems and radio local area networks in the frequency range 900 MHz to 100 GHz," *ITU-R Radiocommunication Sector of ITU*, 2012.

[23] Farwell, M., Ross, J., Luttrell, R., Cohen, D., Chin, W., and Dogaru, T., "Sense through the wall system development and design considerations," *Journal of the Franklin Institute*, Vol. 345, No. 6, 2008, pp. 570 – 591, Advances in Indoor Radar Imaging.

[24] Taylor, C. D., Gutierrez, S. J., Langdon, S. L., Murphy, K. L., and Walton, W. A., *Measurement of RF Propagation into Concrete Structures over the Frequency Range 100 MHZ to 3 GHz*, Springer US, Boston, MA, 1997, pp. 131–144.

[25] Liu, Y. H., Huang, X., Vidojkovic, M., Imamura, K., Harpe, P., Dolmans, G., and Groot, H. D., "A 2.7nJ/b multi-standard 2.3/2.4GHz polar transmitter for wireless sensor networks," *2012 IEEE International Solid-State Circuits Conference*, Feb 2012, pp. 448–450.

[26] Kuo, F. W., Babaie, M., Chen, R., Yen, K., Chien, J. Y., Cho, L., Kuo, F., Jou, C. P., Hsueh, F. L., and Staszewski, R. B., "A fully integrated 28nm Bluetooth Low-Energy transmitter with 36% system efficiency at 3dBm," *European Solid-State Circuits Conference (ESSCIRC), ESSCIRC 2015 - 41st*, Sept 2015, pp. 356–359.

[27] "32.768kHz SMD low profile crystal," .

[28] Shi, Y., Choi, M., Li, Z., Kim, G., Foo, Z., Kim, H. S., Wentzloff, D., and Blaauw, D., "26.7 A 10mm3 syringe-implantable near-field radio system on glass substrate," *2016 IEEE International Solid-State Circuits Conference (ISSCC)*, Jan 2016, pp. 448–449.

[29] Carbonelli, C. and Mengali, U., "M-PPM noncoherent receivers for UWB applications," *IEEE Transactions on Wireless Communications*, Vol. 5, No. 8, Aug 2006, pp. 2285–2294.

[30] Gishkori, S., Leus, G., and Delic, H., "Energy Detection of Wideband and Ultra-Wideband PPM," *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, Dec 2010, pp. 1–5.

[31] Choi, M., Bang, S., Jang, T. K., Blaauw, D., and Sylvester, D., "A 99nW 70.4kHz resistive frequency locking on-chip oscillator with 27.4ppmC temperature stability," *2015 Symposium on VLSI Circuits (VLSI Circuits)*, June 2015, pp. C238–C239.

[32] Wentzloff, D. D., "Ultra-low power radio survey," .

[33] Heide, K., "Convolutional codes," .

[34] Proakis, J. G. and Salehi, M., *Digital communications*, JMcGraw-Hill, 2007.

[35] Jr, G. D. F., "The viterbi algorithm," *Proceedings of the IEEE*, 1973, pp. 268–278.

[36] Atzori, L., Iera, A., and Morabito, G., "The internet of things: A survey," *Computer networks*, Vol. 54, No. 15, 2010, pp. 2787–2805.

[37] Gartner, "Top 10 Strategic Technology Trends for 2014," *URL http://www.gartner.com/newsroom/id/2603623*, October 8,2013.

[38] "MIT Technology Review: The Internet of Things," *URL http://www.technologyreview.com/news/527356/business-adapts-to-a-new-style-of-computer/*, JULY/AUGUST 2014.

[39] "Dynastream Innovations Inc. ANT Message Protocol and Usage," *Application notes URL http://www.thisisant.com*, 2014.

[40] Reiter, G., "Wireless connectivity for the Internet of Things," *URL http://www.ti.com/lit/wp/swry010/swry010.pdf*, 2014.

[41] "IEEE Standard 802.15.4g Part 15.4: Low-Rate Wireless Personal Area Networks - Amendment 3: Physical Layer (PHY) Specifications for Low-Data-Rate Wireless, Smart Metering Utility Networks," April 2012.

[42] van den Heuvel, J., Wu, Y., Baltus, P., Linnartz, J.-P., and van Roermund, A., "Front End Power Dissipation Minimization and Optimal Transmission Rate for Wireless Receivers," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, Vol. 61, No. 5, May 2014, pp. 1566–1577.

[43] Huber, J. and Liu, W., "Data-aided synchronization of coherent CPM receivers," *IEEE Trans. Commun., vol. 40, pp.178 -189*, 1992.

[44] Megali, U. and N.D'Andrea, A., *Synchronization Techniques for Digital Receivers*, SPRINGER SCIENCE+BUSINESS MEDIA,LLC, New York, 1st ed., 1997.

[45] Dutta, D., Karmakar, A., and Saikia, D. K., "Determining Duty Cycle and Beacon Interval with Energy Efficiency and QoS for Low Traffic IEEE 802.15. 4/ZigBee Wireless Sensor Networks," *Advanced Computing, Networking and Informatics-Volume 2*, Springer, 2014, pp. 75–84.

[46] Dally, W. J., Hanrahan, P., Erez, M., Knight, T. J., Labonte, F., Ahn, J.-H., Jayasena, N., Kapasi, U. J., Das, A., Gummaraju, J., and Buck, I., "Merrimac: Supercomputing with Streams," *SC Conference*, Vol. 0, 2003, pp. 35.

[47] Gummaraju, J., Erez, M., Coburn, J., Rosenblum, M., and Dally, W., "Architectural Support for the Stream Execution Model on General-Purpose Processors," *Parallel Architecture and Compilation Techniques, 2007. PACT 2007. 16th International Conference on*, Sept 2007, pp. 3–12.

[48] Seo, S., Dreslinski, R., Who, M., Chakrabarti, C., Mahlke, S., and Mudge, T., "Diet SODA: A power-efficient processor for digital cameras," 2010, pp. 79–84.

[49] Lakshmi, B. and Dhar, A. S., "CORDIC Architectures: A Survey," *VLSI Design, vol. 2010, Article ID 794891, 19 pages, 2010. doi:10.1155/2010/794891*, 2010.

[50] Anderson, J. B., *Digital Transmission Engineering*, Wiley-IEEE Press, 2nd ed., 2005.

[51] Dai, S., Qian, H., Kang, K., and Xiang, W., "A Robust Demodulator for OQP-SKDSSS System," *Circuits, Systems, and Signal Processing*, Vol. 34, No. 1, 2015, pp. 231–247.

[52] Mohammed, K. S., "FPGA Implementation of Bluetooth 2.0 Transceiver," *Proceedings of the 5th WSEAS International Conference on System Science and Simulation in Engineering*, ICOSSE'06, World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, USA, 2006, pp. 295–299.

[53] Wang, A. and Chandrakasan, A., "A 180mV FFT processor using subthreshold circuit techniques," *Solid-State Circuits Conference*, Vol. 1, 2004, pp. 292–529.

[54] Jeon, D., Henry, M., Kim, Y., Lee, I., Zhang, Z., Blaauw, D., and Sylvester, D., "An Energy Efficient Full-Frame Feature Extraction Accelerator With Shift-Latch FIFO in 28 nm CMOS," *Solid-State Circuits, IEEE Journal of*, Vol. 49, No. 5, May 2014, pp. 1271–1284.

[55] Seo, S., Dreslinski, R. G., Woh, M., Park, Y., Charkrabari, C., Mahlke, S., Blaauw, D., and Mudge, T., "Process variation in near-threshold wide SIMD architectures," *Proceedings of the 49th Annual Design Automation Conference*, ACM, 2012, pp. 980–987.

[56] Mikhaylov, K., Plevritakis, N., and Tervonen, J., "Performance Analysis and Comparison of Bluetooth Low Energy with IEEE 802.15.4 and SimpliciTI," *Journal of Sensor and Actuator Networks*, 2013.

[57] Wu, S.-Y., Liaw, J., Lin, C., Chiang, M., Yang, C., Cheng, J., Tsai, M., Liu, M., Wu, P., Chang, C., Hu, L., Lin, C., Chen, H., Chang, S., Wang, S., Tong, P., Hsieh, Y., Pan, K., Hsieh, C., Chen, C., Yao, C., Chen, C., Lee, T., Chang, C., Lin, H., Chen, S., Shieh, J., Tsai, M., Jang, S., Chen, K., Ku, Y., See, Y., and Lo, W., "A highly manufacturable 28nm cmos low power platform technology with fully functional 64mb sram using dual/tripe gate oxide process," *VLSI Technology, 2009 Symposium on*, IEEE, 2009, pp. 210–211.

[58] Dreslinski, R. G., Wieckowski, M., Blaauw, D., Sylvester, D., and Mudge, T., "Near-threshold computing: Reclaiming moore's law through energy efficient integrated circuits," *Proceedings of the IEEE*, Vol. 98, No. 2, 2010, pp. 253–266.

[59] ARM-M0+, "http://www.arm.com/products/processors/cortex-m/cortex-m0plus.php," .

[60] ARM-M3, "http://www.arm.com/products/processors/cortex-m/cortex-m3.php," .

[61] "Nordic Semi. nRF8001 Product Spec.-Bluetooth 4.0," *Journal of Sensor and Actuator Networks*, 2012.

[62] Liu, Y.-H., Huang, X., Vidojkovic, M., Ba, A., Harpe, P., Dolmans, G., and de Groot, H., "A 1.9nJ/b 2.4GHz multistandard (Bluetooth Low Energy/Zigbee/IEEE802.15.6) transceiver for personal/body-area networks," *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2013 IEEE International*, Feb 2013, pp. 446–447.

[63] Retz, G., Shanan, H., Mulvaney, K., Mahony, S., Chanca, M., Crowley, P., Billon, C., Khan, K., and Quinlan, P., "A highly integrated low-power 2.4 GHz transceiver using a direct-conversion diversity receiver in 0.18μm CMOS for IEEE802. 15.4 WPAN," *Solid-State Circuits Conference-Digest of Technical Papers, 2009. ISSCC 2009. IEEE International*, IEEE, 2009, pp. 414–415.

[64] Wong, A., Dawkins, M., Devita, G., Kasparidis, N., Katsiamis, A., King, O., Lauria, F., Schiff, J., and Burdett, A., "A 1V 5mA multimode IEEE 802.15.6/bluetooth low-energy WBAN transceiver for biotelemetry applications," *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2012 IEEE International*, Feb 2012, pp. 300–302.

[65] Casilari, E., Cano-García, J. M., and Campos-Garrido, G., "Modeling of current consumption in 802.15. 4/ZigBee sensor motes," *Sensors*, Vol. 10, No. 6, 2010, pp. 5443–5468.

[66] Fourty, N., van den Bossche, A., and Val, T., "An advanced study of energy consumption in an {IEEE} 802.15.4 based network: Everything but the truth on 802.15.4 node lifetime," *Computer Communications*, Vol. 35, No. 14, 2012, pp. 1759 – 1767, Special issue: Wireless Green Communications and Networking.

[67] Dementyev, A., Hodges, S., Taylor, S., and Smith, J., "Power consumption analysis of Bluetooth Low Energy, ZigBee and ANT sensor nodes in a cyclic sleep scenario," *Wireless Symposium (IWS), 2013 IEEE International*, April 2013, pp. 1–4.

[68] Lin, Y., Lee, H., Woh, M., Harel, Y., Mahlke, S., Mudge, T., Chakrabarti, C., and Flautner, K., "SODA: A Low-power Architecture For Software Radio," *Proceedings of the 33rd Annual International Symposium on Computer Architecture*, ISCA '06, IEEE Computer Society, Washington, DC, USA, 2006, pp. 89–101.

[69] Woh, M., Seo, S., Mahlke, S., Mudge, T., Chakrabarti, C., and Flautner, K., "AnySP: Anytime Anywhere Anyway Signal Processing," *Proceedings of the 36th Annual International Symposium on Computer Architecture*, ISCA '09, ACM, New York, NY, USA, 2009, pp. 128–139.

[70] Woh, M., Lin, Y., Seo, S., Mahlke, S., Mudge, T., Chakrabarti, C., Bruce, R., Kershaw, D., Reid, A., Wilder, M., and Flautner, K., "From SODA to scotch: The evolution of a wireless baseband processor," *Microarchitecture, 2008. MICRO-41. 2008 41st IEEE/ACM International Symposium on*, Nov 2008, pp. 152–163.

[71] Chen, Y., Lu, S., Kim, H. S., Blaauw, D., Dreslinski, R. G., and Mudge, T., "A low power software-defined-radio baseband processor for the Internet of Things,"

2016 IEEE International Symposium on High Performance Computer Architecture (HPCA), March 2016, pp. 40–51.

[72] Chen, Y., Chiotellis, N., Chuo, L. X., Pfeiffer, C., Shi, Y., Dreslinski, R. G., Grbic, A., Mudge, T., Wentzloff, D. D., Blaauw, D., and Kim, H. S., "Energy-Autonomous Wireless Communication for Millimeter-Scale Internet-of-Things Sensor Nodes," *IEEE Journal on Selected Areas in Communications*, Vol. PP, No. 99, 2016, pp. 1–1.

[73] Kim, H. S. and Daneshrad, B., "Energy-Constrained Link Adaptation for MIMO OFDM Wireless Communication Systems," *IEEE Transactions on Wireless Communications*, Vol. 9, No. 9, September 2010, pp. 2820–2832.

[74] El-Rayis, A. O., Zhao, X., Arslan, T., and Erdogan, A. T., "Dynamically programmable Reed Solomon processor with embedded Galois Field multiplier," *ICECE Technology, 2008. FPT 2008. International Conference on*, Dec 2008, pp. 269–272.

[75] Howard, S. L., Schlegel, C., Iniewski, K., and Iniewski, K., "Error Control Coding in Low-Power Wireless Sensor Networks: When Is ECC Energy-Efficient?" *EURASIP Journal on Wireless Communications and Networking*, Vol. 2006, No. 1, 2006, pp. 074812.

[76] Kapoor, V., Abraham, V. S., and Singh, R., "Elliptic Curve Cryptography," *Ubiquity*, Vol. 2008, No. May, May 2008, pp. 7:1–7:8.

[77] Lu, S., Zhang, Z., and Papaefthymiou, M., "1.32GHz high-throughput charge-recovery AES core with resistance to DPA attacks," *2015 Symposium on VLSI Circuits (VLSI Circuits)*, June 2015, pp. C246–C247.

[78] Lin, Y. M., Chang, H. C., and Lee, C. Y., "Improved High Code-Rate Soft BCH Decoder Architectures With One Extra Error Compensation," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 21, No. 11, Nov 2013, pp. 2160–2164.

[79] Lu, Y.-K., Shieh, M.-D., and Kuo, W.-H., "Design of high-speed errors-and-erasures Reed-Solomon decoders for multi-mode applications," *VLSI Design, Automation and Test, 2009. VLSI-DAT'09. International Symposium on*, IEEE, 2009, pp. 199–202.

[80] Wicker, S. B., *Error control systems for digital communication and storage*, Vol. 1, Prentice hall Englewood Cliffs, 1995.

[81] Mozhiarasi, P., Gayathri, C., and Deepan, V., "An Enhanced (31, 11, 5) Binary BCH Encoder and Decoder for Data Transmission," .

[82] KIM, H., Lee, S., and Goel, M., "Method, device, and digital circuitry for providing a closed-form solution to a scaled error locator polynomial used in BCH decoding," March 5 2013, US Patent 8,392,806.

[83] Mathew, P., Augustine, L., G., S., and Devis, T., "Hardware Implementation of (63, 51) BCH Encoder and Decoder For WBAN Using LFSR and BMA," *CoRR*, Vol. abs/1408.2908, 2014.

[84] Qian, H., Dai, S., Kang, K., and Wang, X., "Efficient coding schemes for low-rate wireless personal area networks," *IET Communications*, Vol. 10, No. 8, 2016, pp. 915–921.

[85] Rosas, F., Brante, G., Souza, R. D., and Oberli, C., "Optimizing the code rate for achieving energy-efficient wireless communications," *2014 IEEE Wireless Communications and Networking Conference (WCNC)*, April 2014, pp. 775–780.

[86] Kumar, A. and van Berkel, K., "Vectorization of Reed Solomon decoding and mapping on the EVP," *Proceedings of the conference on Design, automation and test in Europe*, ACM, 2008, pp. 450–455.

[87] Deschamps, J.-P., *Hardware implementation of finite-field arithmetic*, McGraw-Hill, Inc., 2009.

[88] Canright, D., "A Very Compact S-box for AES," *Proceedings of the 7th International Conference on Cryptographic Hardware and Embedded Systems*, CHES'05, Springer-Verlag, Berlin, Heidelberg, 2005, pp. 441–455.

[89] Lim, C. H. and Hwang, H. S., *Speeding Up Elliptic Scalar Multiplication with Precomputation*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2000, pp. 102–119.

[90] López, J. and Dahab, R., *Improved Algorithms for Elliptic Curve Arithmetic in GF(2n)*, Springer Berlin Heidelberg, Berlin, Heidelberg, 1999, pp. 201–212.

[91] Brown, D., "SEC 2: Recommended Elliptic Curve Domain Parameters," 2010.

[92] of Standards, N. I. and Technology, "Digital Signature Standard, FIPS Publication," Feb 2000.

[93] Stein, Y. and Kablotsky, J., "Compact Galois field multiplier engine," Feb. 13 2007, US Patent 7,177,891.

[94] Stein, Y., Primo, H., and Kablotsky, J., "Galois field multiplier system," July 20 2004, US Patent 6,766,345.

[95] Jain, S. K., Song, L., and Parhi, K. K., "Efficient semisystolic architectures for finite-field arithmetic," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 6, No. 1, March 1998, pp. 101–113.

[96] Pan, J. S., Lee, C. Y., and Meher, P. K., "Low-Latency Digit-Serial and Digit-Parallel Systolic Multipliers for Large Binary Extension Fields," *IEEE Transactions on Circuits and Systems I: Regular Papers*, Vol. 60, No. 12, Dec 2013, pp. 3195–3204.

[97] Ibrahim, A. and Gebali, F., "Low Power Semi-systolic Architectures for Polynomial-Basis Multiplication over GF(2 m) Using Progressive Multiplier Reduction," *Journal of Signal Processing Systems*, Vol. 82, No. 3, 2016, pp. 331–343.

[98] Guajardo, J., *Itoh–Tsujii Inversion Algorithm*, Springer US, Boston, MA, 2011, pp. 650–653.

[99] Wu, C.-H., Wu, C.-M., Shieh, M.-D., and Hwang, Y.-T., "High-speed, low-complexity systolic designs of novel iterative division algorithms in GF(2m)," *IEEE Transactions on Computers*, Vol. 53, No. 3, Mar 2004, pp. 375–380.

[100] Kobayashi, K. and Takagi, N., "Fast Hardware Algorithm for Division in GF(2m) Based on the Extended Euclid's Algorithm With Parallelization of Modular Reductions," *IEEE Transactions on Circuits and Systems II: Express Briefs*, Vol. 56, No. 8, Aug 2009, pp. 644–648.

[101] Lin, W. C., Shieh, M. D., and Wu, C. M., "Design of high-speed bit-serial divider in GF(2m)," *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, May 2010, pp. 713–716.

[102] Luo, J., Bowers, K. D., Oprea, A., and Xu, L., "Efficient Software Implementations of Large Finite Fields GF(2N) for Secure Storage Applications," *Trans. Storage*, Vol. 8, No. 1, Feb. 2012, pp. 2:1–2:27.

[103] "TI opensource AES: http://www.ti.com/tool/AES-128," .

[104] "NXP AES benchmarks: https://www.lpcware.com/content/project/software-encryption-nxp-arm-microcontrollers," .

[105] "ARM AES benchmarks: https://www.ietf.org/proceedings/92/slides/slides-92-lwig-3.pdf," .

[106] Seo, H. and Kim, H., "Optimized multi-precision multiplication for public-key cryptography on embedded microprocessors," *International Journal of Computer and Communication Engineering*, Vol. 2, No. 3, 2013, pp. 255.

[107] Hutter, M. and Wenger, E., "Fast Multi-precision Multiplication for Public-key Cryptography on Embedded Microprocessors," *Proceedings of the 13th International Conference on Cryptographic Hardware and Embedded Systems*, CHES'11, Springer-Verlag, Berlin, Heidelberg, 2011, pp. 459–474.

[108] de Clercq, P., "Public Key Cryptography in 32-bit for Ultra Low-Power Microcontrollers," 2014.

[109] de Clercq, R., Uhsadel, L., Van Herrewege, A., and Verbauwhede, I., "Ultra Low-Power Implementation of ECC on the ARM Cortex-M0+," *Proceedings of the 51st Annual Design Automation Conference*, DAC '14, ACM, New York, NY, USA, 2014, pp. 112:1–112:6.

[110] Erdem, S. S., "Fast software multiplication in F_2 [x] for embedded processors," *Turkish Journal of Electrical Engineering & Computer Sciences*, Vol. 20, No. 4, 2012, pp. 593–605.

[111] Karatsuba, A. and Ofman, Y., "Multiplication of Multidigit Numbers on Automata," *Soviet Physics–Doklady*, Vol. 7, 1963, pp. 595–596.

[112] Lederer, C., Mader, R., Koschuch, M., Großschädl, J., Szekely, A., and Tillich, S., *Energy-Efficient Implementation of ECDH Key Exchange for Wireless Sensor Networks*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 112–127.

[113] Jeon, D., Henry, M. B., Kim, Y., Lee, I., Zhang, Z., Blaauw, D., and Sylvester, D., "An Energy Efficient Full-Frame Feature Extraction Accelerator With Shift-Latch FIFO in 28 nm CMOS," *IEEE Journal of Solid-State Circuits*, Vol. 49, No. 5, May 2014, pp. 1271–1284.

[114] Mathew, S., Satpathy, S., Suresh, V., Anders, M., Kaul, H., Agarwal, A., Hsu, S., Chen, G., and Krishnamurthy, R., "340 mV-1.1 V, 289 Gbps/W, 2090-Gate NanoAES Hardware Accelerator With Area-Optimized Encrypt/Decrypt GF(2 4) 2 Polynomials in 22 nm Tri-Gate CMOS," *IEEE Journal of Solid-State Circuits*, Vol. 50, No. 4, April 2015, pp. 1048–1058.

[115] Zhang, Y., Yang, K., Saligane, M., Blaauw, D., and Sylvester, D., "A compact 446 Gbps/W AES accelerator for mobile SoC and IoT in 40nm," *2016 IEEE Symposium on VLSI Circuits (VLSI-Circuits)*, June 2016, pp. 1–2.

[116] Lee, J.-W., Chen, Y.-L., Tseng, C.-Y., Chang, H.-C., and Lee, C.-Y., "A 521-bit dual-field elliptic curve cryptographic processor with power analysis resistance," *ESSCIRC, 2010 Proceedings of the*, IEEE, 2010, pp. 206–209.

[117] Mathew, S., Kounavis, M., Sheikh, F., Hsu, S., Agarwal, A., Kaul, H., Anders, M., Berry, F., and Krishnamurthy, R., "3GHz, 74mW 2-level Karatsuba 64b Galois field multiplier for public-key encryption acceleration in 45nm CMOS," *ESSCIRC, 2010 Proceedings of the*, Sept 2010, pp. 198–201.

[118] Sayilar, G. and Chiou, D., "Cryptoraptor: High throughput reconfigurable cryptographic processor," *2014 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, Nov 2014, pp. 155–161.

[119] Hutter, M., Feldhofer, M., and Wolkerstorfer, J., "A Cryptographic Processor for Low-resource Devices: Canning ECDSA and AES Like Sardines," *Proceedings of the 5th IFIP WG 11.2 International Conference on Information Security Theory and Practice: Security and Privacy of Mobile Devices in Wireless Communication*, WISTP'11, Springer-Verlag, Berlin, Heidelberg, 2011, pp. 144–159.

[120] Atieno, L., Allen, J., Goeckel, D., and Tessier, R., "An adaptive Reed-Solomon errors-and-erasures decoder," *Proceedings of the 2006 ACM/SIGDA 14th international symposium on Field programmable gate arrays*, ACM, 2006, pp. 150–158.

[121] Ruiz, C. E. S., "RS Decoder (255, k) in Reconfigurable Hardware Oriented Towards Cognitive Radio," *Ingenieria y Universidad*, Vol. 17, No. 1, 2013, pp. 77–92.

[122] Lu, Y. K. and Shieh, M. D., "Efficient architecture for Reed-Solomon decoder," *VLSI Design, Automation, and Test (VLSI-DAT), 2012 International Symposium on*, April 2012, pp. 1–4.

[123] Targhetta, A. D., Owen, D. E., Israel, F. L., and Gratz, P. V., "Energy-efficient implementations of GF (p) and GF(2m) elliptic curve cryptography," *2015 33rd IEEE International Conference on Computer Design (ICCD)*, Oct 2015, pp. 704–711.