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

Yajing Chen, Shengshuo Lu, Hun-Seok Kim, David Blaauw, Ronald G. Dreslinski, Trevor Mudge
University of Michigan, Ann Arbor

ABSTRACT

In this paper, 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× 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 configurable 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 consumption, 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.

1. INTRODUCTION

The Internet of Things (IoT) [1] 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 [2]. And the number of “things" that will be connecting to the Internet will exceed PCs and smartphones [3].

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.

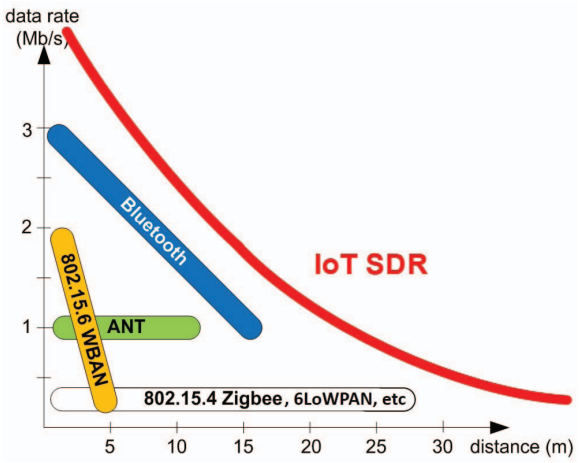

Figure 1: The Application Domain of the IoT SDR. IoT standards support either long distance or high data rate to limit power consumption. An **IoT SDR** can be software defined to operate at points in the region under the red line.

Several wireless standards have been proposed for the IoT communications. Figure 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 1 shows.

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 1. IEEE 802.15.4 [4] 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

978-1-4673-9211-2/16/$31.00

communications.

Bluetooth [5] 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 [6] 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 [7] 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 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 [8]. 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 non-standard protocols on the baseband processing, an SDR architecture will bring more degrees of freedom in the space of data rate, distance and energy 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.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 the power dissipation over air follows an inverse square law, which cannot be avoided. 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.

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 [5], ANT [6] and 802.15.4g FSK-SUN [9] all employ FSK modulation. 802.15.6WBAN [7] and 802.15.4-OQPSK [4] 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.

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 paper, 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 paper 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 tradeoffs 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.

2. BASEBAND PROCESSING OF IOT COMMUNICATIONS

The baseband processing at the receiver is more complicated than at the transmitter [10] for two reasons: the un-

certainty 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 2 illustrates a general communications system.

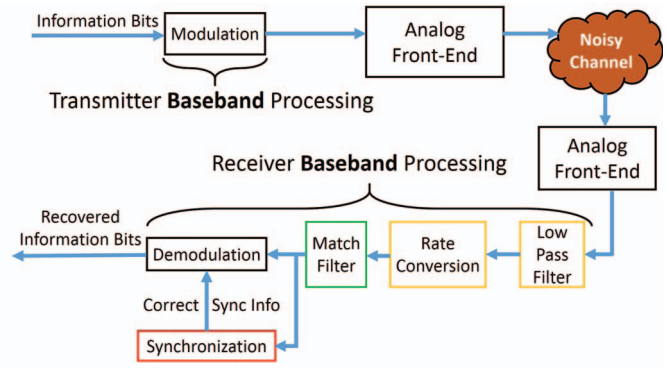

Figure 2: Baseband Processing of an IoT Communications System. The receiver is more complicated than the transmitter. It adds a Low Pass Filter, Matched Filter and Synchronization which are computationally intensive.

2.1 FIR Filter Computation Characteristics

There are two basic FIR filters in IoT communications systems shown in Figure 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.

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

Three observations can be made from eq. (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.

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 [11] 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. (2).

$$\tau = argmin_t \left\| \underline{r}[t] - \underline{p} \right\| \tag{2}$$
$$\text{where } \underline{r}[t] = [r[t], r[t+1], ..., r[t+P-1]]$$
$$\underline{p} = [p_0, p_1, p_2, ..., p_{P-1}]$$

The received vector at time t is $\underline{r}[t]$, and $\underline{p}$ is the reference signal vector. P is the vector length. Eq. (2) can also be reduced to eq. (3), where p_j^* is the conjugate of p_j.

$$\tau = argmax_t \left| \sum_{j=0}^{P-1} r[t+j] p_j^* \right| \tag{3}$$

Eq. (3) is the well-known Maximum Likelihood (ML) timing estimation [12]. Eq. (2) can be written as eq. (4) when both $\underline{r}[t]$ and $\underline{p}$ are real-number vectors.

$$\tau = \underset{t}{argmin} |\underline{r}[t] - p| = argmin_t \sum_{j=0}^{P-1} \left| r[t+j] - p_j \right| \tag{4}$$

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) and eq. (4) is the computationally intensive operation since τ needs to be reevaluated for every incoming sample.

Similarly to FIR filters, both eq. (3) and eq. (4) are streaming in nature, vectorizable, and require a reduction operation. In addition, max and min operations are required which can be handled in a non-vector unit.

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× 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.

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 2 can be merged into one step by employing Streaming Registers, discussed in Section 3.2.1. In this way, the computation 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.

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 [13]. 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.

2.6 Communications Datapath Bit Width

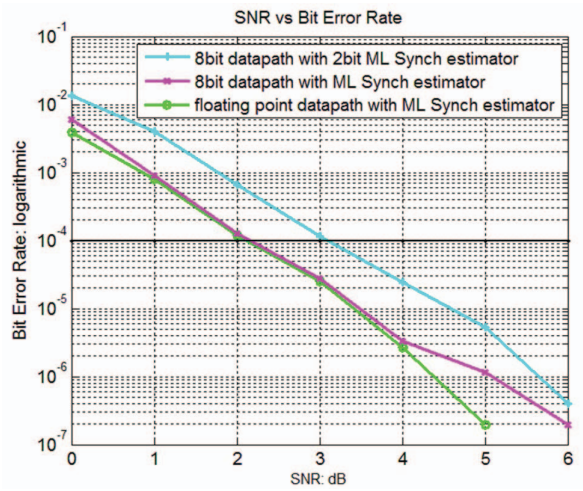

Figure 3: Signal-to-Noise Ratio (SNR) vs Bit Error Rate (BER) for Different Bit Width Implementation. The 8-bit communications datapath captures almost the same precision as the floating point datapath does. The 2-bit Synchronization estimator is sufficient to achieve the target BER.

We conduct system simulations with different bit width datapaths. The influence of bit width is shown in Figure 3. 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. 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 [4], 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 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), 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.

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 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. 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.

3.1 System Architecture

The system architecture is presented in Figure 4. 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) and eq. (4) of the synchronization stage.

3.2 Microarchitecture

The microarchitecture of this IoT SDR datapath is illustrated in Figure 5. There are two computational units: the SIMD Unit and the Scalar Unit.

3.2.1 SIMD Unit

The SIMD Unit is responsible for computationally intensive vector and reduction operations. There are a total of

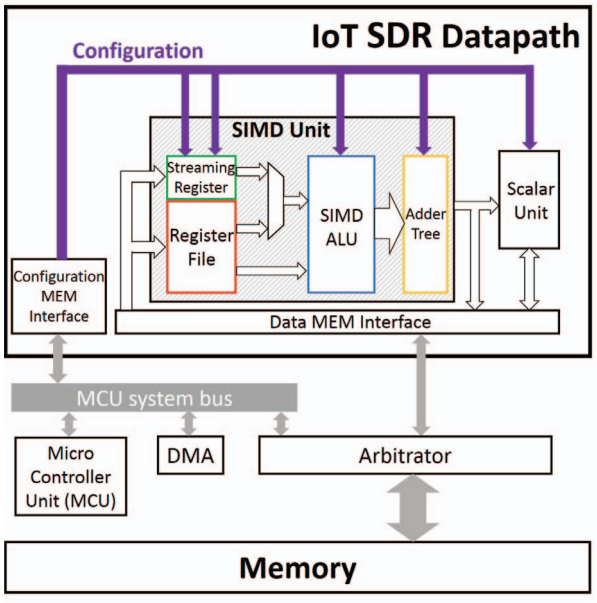

Figure 4: IoT SDR System Architecture. The IoT SDR is controlled by an MCU; data is shared by the MCU and IoT SDR via memory.

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 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. (1), eq. (3) and eq. (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.

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=4×4). 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. During an IDLE configuration, neither the streaming registers nor the regular register file has switching 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 [14, 15] 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 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× 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×** 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

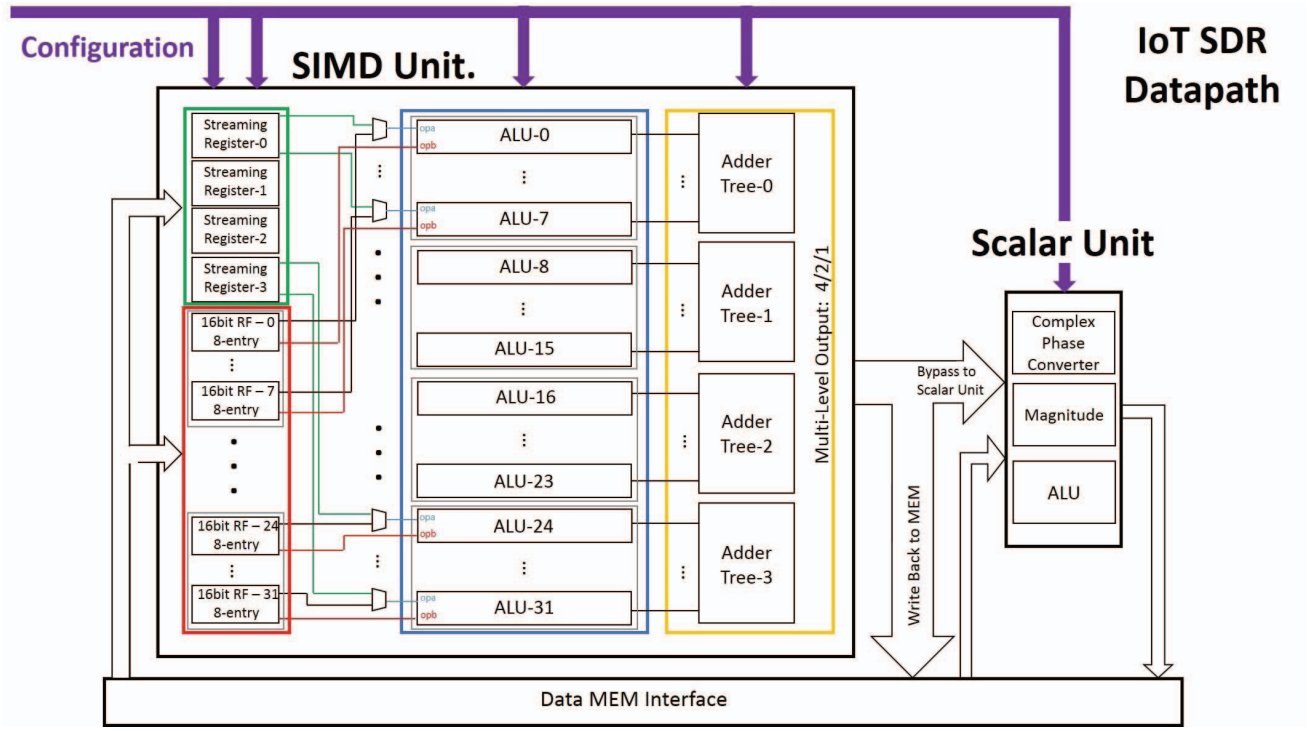

Figure 5: The Microarchitecture of the IoT SDR Datapath. An IoT SDR datapath comprises one **Scalar Unit** and one **SIMD Unit** consisting of Streaming Registers, a Register File, SIMD-ALUs and an Adder Tree.

ALU. Each lane will take two 16-bit operands and provide 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** is the reduction unit that is introduced to accelerate large summations. More features are used compared to previous adder-trees [16]. 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.

3.2.2 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 5. The Complex Phase Converter [17] 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.

3.2.3 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.

4. EVALUATION METHOD

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.

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 1. 802.15.4-OQPSK represents standards based on quadrature linear modulation schemes while BLE demonstrates filtered Frequency Shift Keying 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 handcoded 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) [18]. 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 [19]. The communications datapath of 802.15.4-OQPSK receiver is shown in Figure 6. 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.

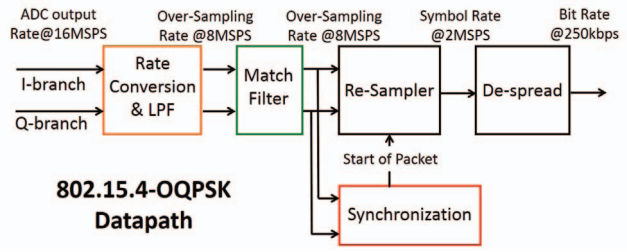

Figure 6: 802.15.4-OQPSK Receiver Communications Datapath. The modulation scheme is **O**ffset-**Q**uadrature **P**hase **S**hift **K**eying. The over-sampling rate is **8MSPS**.

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 7. 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) [18], 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 [20].The over-sampling rate of Bluetooth at 1Mbps is 4M real samples per second.

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.

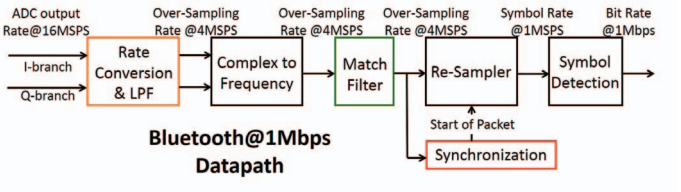

Figure 7: BLE Receiver Communications Datapath. The modulation scheme is **GFSK**.The over-sampling rate is **4MSPS**

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 1 lists the computational characteristics of each kernel in the receiver communications datapaths of 802.15.4-OQPSK and BLE.

4.4 Power and Area Estimation Methodology

In this section, the methodology to estimate the power and area of the IoT SDR design is presented.

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 [4, 5]. 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

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
Match Filter	8	8-bit/samp complex	8MSPS	SIMD Unit
Synchronization*	128	4-bit/samp complex	8MSPS	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	4MSPS	Scalar Unit (Complex to Phase)
Match Filter	16	8-bit/samp real	4MSPS	SIMD Unit
Synchronization*	128	8-bit/samp real	4MSPS	SIMD Unit (subabs) Scalar Unit (Magnitude)
Re-sample	NaN	8-bit/samp real	1MSPS	Scalar Unit
Symbol detection	NaN	8-bit/samp real	1MSPS	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

Table 1. Computational Characteristics and Kernel Mappings for the two full receiver systems.

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 simulation. 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.

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 paper, the **peak power** of the Synchronization mode is used to characterize the IoT SDR architecture.

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 8, both SPICE simulation and PrimeTime/Power simulation are applied to the representative circuits at 1V/0.9V/0.8V. Figure 8 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 9 shows the voltage scaling trend of the entire IoT SDR baseband datapath, in terms of energy/cycle and critical path delay.

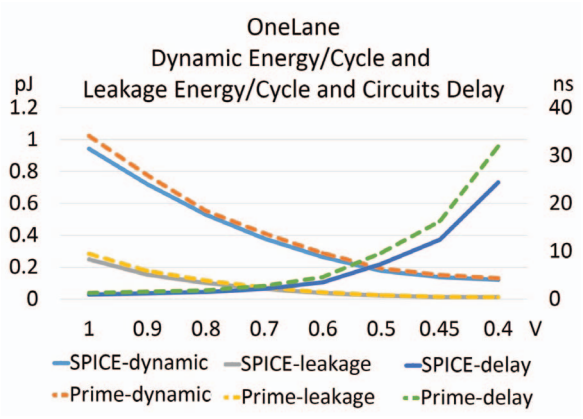

Figure 8: Voltage Scaling Trend of Representative Circuits. Voltage Scaling is used to meet the power constraint. The Voltage Scaling Ratio is acquired from SPICE.

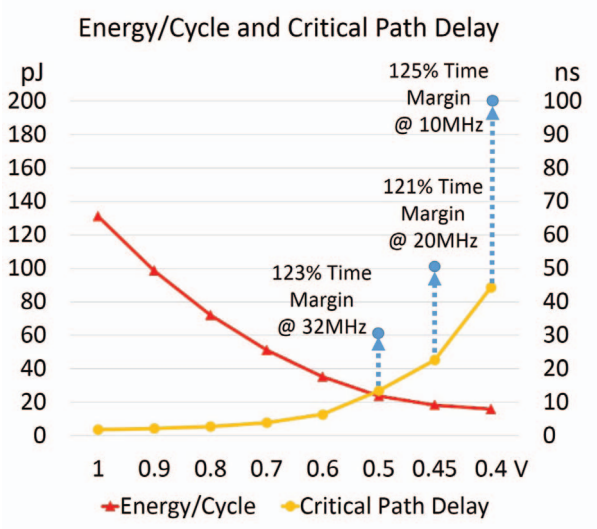

Figure 9: Voltage Scaling Trend of the IoT SDR baseband datapath. A more than **100%** margin is reserved at each operating points for reliable operation.

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 [21].

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 [22] 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, et al. [23] for wide SIMD processors.

The SRAM and MCU are operated at nominal voltage.

5.2 Power Breakdown

Figure 10 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.

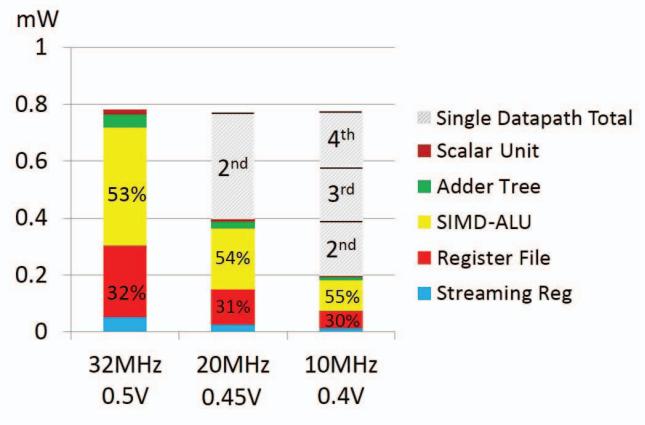

Figure 10: Power Breakdown. The SIMD-ALU and Register File dominate the power at all frequency settings.

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 10 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 10 is more than 10Top/J, which is very efficient and comparable to former ASIC designs in 28nm [22].

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 11 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.

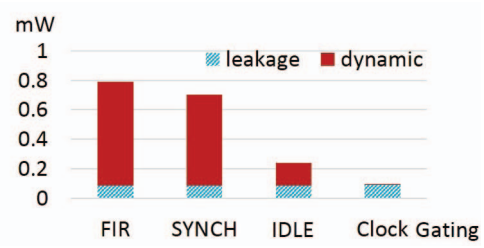

Figure 11: Kernel Based Power Consumption.

5.4 Area

The layout of the IoT SDR datapath is shown in Figure 12. Its area is 0.074mm^2.

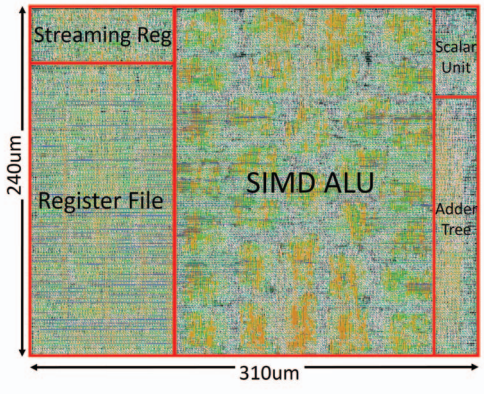

Figure 12: Layout

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.

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 2, depends on the number of shared kernels and their computational demands. By

1× IoT SDR	2× IoT SDR	4× IoT SDR
Time Multiplexing		No Time Multiplexing
IoT SDR Datapath Time Multiplexing by *Rate Conversion & LPF, Match Filter, Synchronization-Step0, Synchronization-Step1*	IoT SDR Datapath 0 Time Multiplexing by *Rate Conversion & LPF, Match Filter*	IoT SDR Datapath 0 *Rate Conversion & LPF*
		IoT SDR Datapath 1 *Match Filter*
	IoT SDR Datapath 1 Time Multiplexed by *Synchronization-Step0, Synchronization-Step1*	IoT SDR Datapath 2 *Synchronization- Step0*
		IoT SDR Datapath 3 *Synchronization-Step1*
Freq > 8M + 8M/4 + 2×8M **	Freq0 > 8M + 8M/4 ** Freq1 > 8M + 8M **	Freq0 > 8M ** Freq1 > 8M/4 ** Freq2 > 8M ** Freq3 > 8M **
1× IoT SDR Datapath @ Freq = 32MHz ***	2× IoT SDR Datapath @ Freq = 20MHz ***	4× IoT SDR Datapath @Freq = 10MHz ***

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

Table 2. System Mapping.

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

* Frequency(IoT Datapath) must be greater than Sample Frequency
** No memory for IoT Datapath Memory is only for M0+

Table 3. Receiver System Configurations. MCU and Memory operate at nominal voltage

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 [24]. For the system configurations in Table 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 buffers are added to avoid unnecessary memory accesses. For the method, "4× IoT SDR Datapath" in Table 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 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 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 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 10. 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. [25], while the power of the memory is scaled down to the equivalent value in a 28nm process [25]. We run the memory at the nominal voltage to avoid any reliability issues [26].

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 [27] and scales from 40nm to 28nm according to [28]. As shown in Table 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.

6.2 System Power and Area

The total system power and area are listed in Table 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.

Figure 13 illustrates the power/area trade-off for three system configurations. The system power and area are pre-

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	0.014
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**

Table 4. System Power and Area

sented 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.

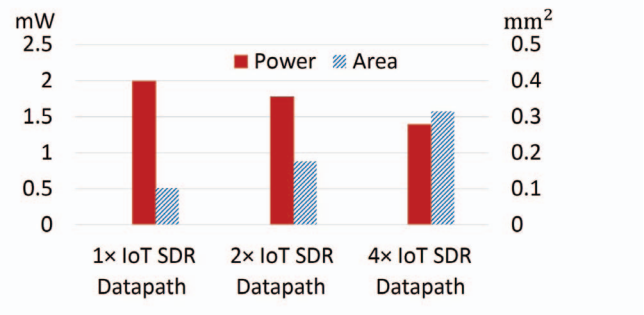

Figure 13: System Power and Area Trade-offs.

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×, 4.7× and 1.46× individually, resulting in a total of a 71× 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 [29]. 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 [30, 31, 32], focus on RF/analog and only include small portion of digital baseband processing. Sensor network research [33, 34, 35] targets for topics such as network layer processing, even they measure the power/energy consumption of the sensor nodes, utilizing commercial transceivers.

7. RELATED WORK

Previous SDR work, such as SODA [36], AnySP [37] and Ardbeg [38], 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.

8. CONCLUSION

In this paper, 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× 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.

Acknowledgement This research is supported by ARM Ltd, and a grant from NSF.

9. REFERENCES

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

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

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

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

[5] "Bluetooth SIG. Bluetooth specification version 4.0," in *The Bluetooth Special Interest Group*, 30 June 2010.

[6] "Dynastream Innovations Inc. ANT message protocol and usage," in *Application notes URL http://www.thisisant.com*, 2014.

[7] "IEEE Standard 802.15.6 part 15.6: Wireless body area networks," 29 February 2012.

[8] G. Reiter, "Wireless connectivity for the internet of things," in *URL http://www.ti.com/lit/wp/swry010/swry010.pdf*, 2014.

[9] "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.

[10] J. van den Heuvel, Y. Wu, P. Baltus, J.-P. Linnartz, and A. van Roermund, "Front end power dissipation minimization and optimal transmission rate for wireless receivers," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 61, pp. 1566–1577, May 2014.

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

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

[13] D. Dutta, A. Karmakar, and D. K. Saikia, "Determining duty cycle and beacon interval with energy efficiency and qos for low traffic ieee 802.15. 4/zigbee wireless sensor networks," in *Advanced Computing, Networking and Informatics-Volume 2*, pp. 75–84, Springer, 2014.

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

[15] J. Gummaraju, M. Erez, J. Coburn, M. Rosenblum, and W. Dally, "Architectural support for the stream execution model on general-purpose processors," in *Parallel Architecture and Compilation Techniques, 2007. PACT 2007. 16th International Conference on*, pp. 3–12, Sept 2007.

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

[17] B. Lakshmi and A. S. Dhar, "Cordic architectures: A survey," in *VLSI Design, vol. 2010, Article ID 794891, 19 pages, 2010. doi:10.1155/2010/794891*, 2010.

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

[19] S. Dai, H. Qian, K. Kang, and W. Xiang, "A robust demodulator for oqpskâĂŞdsss system," *Circuits, Systems, and Signal Processing*, vol. 34, no. 1, pp. 231–247, 2015.

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

[21] A. Wang and A. Chandrakasan, "A 180mv fft processor using subthreshold circuit techniques," in *Solid-State Circuits Conference*, vol. 1, pp. 292–529, 2004.

[22] D. Jeon, M. Henry, Y. Kim, I. Lee, Z. Zhang, D. Blaauw, and D. Sylvester, "An energy efficient full-frame feature extraction accelerator with shift-latch fifo in 28 nm cmos," *Solid-State Circuits, IEEE Journal of*, vol. 49, pp. 1271–1284, May 2014.

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

[24] K. Mikhaylov, N. Plevritakis, and J. Tervonen, "Performance analysis and comparison of bluetooth low energy with ieee 802.15.4 and simpliciti," in *Journal of Sensor and Actuator Networks*, 2013.

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

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

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

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

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

[30] Y.-H. Liu, X. Huang, M. Vidojkovic, A. Ba, P. Harpe, G. Dolmans, and H. de Groot, "A 1.9nj/b 2.4ghz multistandard (bluetooth low energy/zigbee/ieee802.15.6) transceiver for personal/body-area networks," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2013 IEEE International*, pp. 446–447, Feb 2013.

[31] G. Retz, H. Shanan, K. Mulvaney, S. Mahony, M. Chanca, P. Crowley, C. Billon, K. Khan, and P. Quinlan, "A highly integrated low-power 2.4 ghz transceiver using a direct-conversion diversity receiver in 0.18μm cmos for ieee802. 15.4 wpan," in *Solid-State Circuits Conference-Digest of Technical Papers, 2009. ISSCC 2009. IEEE International*, pp. 414–415, IEEE, 2009.

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

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

[34] N. Fourty, A. van den Bossche, and T. Val, "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, pp. 1759 – 1767, 2012. Special issue: Wireless Green Communications and Networking.

[35] A. Dementyev, S. Hodges, S. Taylor, and J. Smith, "Power consumption analysis of bluetooth low energy, zigbee and ant sensor nodes in a cyclic sleep scenario," in *Wireless Symposium (IWS), 2013 IEEE International*, pp. 1–4, April 2013.

[36] Y. Lin, H. Lee, M. Woh, Y. Harel, S. Mahlke, T. Mudge, C. Chakrabarti, and K. Flautner, "Soda: A low-power architecture for software radio," in *Proceedings of the 33rd Annual International Symposium on Computer Architecture*, ISCA '06, (Washington, DC, USA), pp. 89–101, IEEE Computer Society, 2006.

[37] M. Woh, S. Seo, S. Mahlke, T. Mudge, C. Chakrabarti, and K. Flautner, "Anysp: Anytime anywhere anyway signal processing," in *Proceedings of the 36th Annual International Symposium on Computer Architecture*, ISCA '09, (New York, NY, USA), pp. 128–139, ACM, 2009.

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