A Two-Tiered Heterogeneous and Reconfigurable Application Processor for Future Internet of Things

Prasanna Kansakar* and Arslan Munir†
Department of Computer Science
Kansas State University, Manhattan, KS, USA
Email: *pkansakar@ksu.edu, †amunir@ksu.edu

***Abstract*—The Internet of things (IoT) is leading the world into a future of ubiquitous connectivity. The heterogeneity within the IoT domain necessitates a highly flexible, secure, dependable, and energy-efficient IoT processor architecture. In this paper, we propose a novel processor architecture for IoT that renders energy efficiency, high-performance, flexibility, security, and dependability to meet the diverse application requirements. To address the stringent *energy efficiency* demands of IoT devices, we propose a two-tiered heterogeneous processor architecture that is composed of a *high-performance* optimized reconfigurable host processor which controls a number of low-power optimized interface processors. The proposed IoT architecture also incorporates reconfigurability in host processors' computing and communication parameters and co-processor extensions to impart *flexibility* and additional energy savings. The proposed IoT architecture contains various *security* co-processor extensions to support various security primitives including encryption and decryption, key generation, integrity verification, and device authentication. Finally, the proposed architecture incorporates reliability and *dependability* through various hardware- and software-based fault tolerance methods. Experimental results present and compare microarchitecture configurations for host and interface processors obtained through an efficient design space exploration methodology. We have implemented selected security and dependability primitives of our proposed IoT architecture on a Xilinx Spartan-6 field-programmable gate array (FPGA). Results reveal that our proposed IoT architecture can attain a speedup of 47.93× while consuming 2.4× lesser energy for furnishing security and dependability primitives as compared to an optimized ARM implementation of similar security and dependability primitives.**

***Index Terms*—Internet of things (IoT), reconfigurable processor, heterogeneous processor, microarchitecture, security, dependability, energy efficiency, fault tolerance**

I. Introduction and Motivation

The Internet of things (IoT) is a new paradigm in computing wherein everyday physical objects are interconnected through an intelligent, invisible network fabric which allows for objects in the IoT ecosystem to communicate, directly or indirectly, with each other or the Internet for purposes of automation, remote data sensing, and centralized management/control [1]. There are two approaches being considered for IoT deployments. The first approach involves deploying IoT devices as "dumb nodes" with limited processing and communication capabilities. In this approach, the bulk of the data processing and analysis is carried out in the computing nodes higher up in the network hierarchy. The second approach involves incorporating higher processing and communication capabilities in the IoT devices such that only minimal access to computing nodes higher up in the network hierarchy is required. This approach provides a balance between how much computation needs to be done locally versus how much computation needs to be done globally by considering the tradeoffs between monetary cost, real-time responsiveness, energy efficiency, network latency, and network congestion [2].

In order to make a choice between the above mentioned two IoT deployment approaches, several IoT-specific constraints have to be considered such as cost, performance, energy efficiency, etc. Specifically, IoT devices are mostly battery powered and thus are highly energy constrained. Some devices must operate throughout their entire lifetime on the battery they are deployed with whereas other devices may have limited charging mechanisms. All processing and communication activities should therefore be highly energy-efficient. An always active processor is not a viable implementation for IoT devices. Energy efficiency can be achieved in two ways in IoT architectures. The first way is to deploy IoT devices with single high-performance processor with energy conserving sleep modes and the second way is to create a heterogeneous architecture that consists of a network of low-power processors governed by a high-performance processor. Out of these two ways of achieving energy efficiency, the second approach with the heterogeneous architecture is more promising because in the first implementation, the energy required for waking up a high-performance processor from sleep mode is high [3]. Hence, the first implementation, although better than an always active processor implementation, is still relatively energy-inefficient [3]. The heterogeneous architecture has been adopted by ARM and Synopsys for designing their energy-efficient IoT solutions [3] [4].

Another key constraint that needs to be considered when developing an IoT architecture is the need for interoperability. The IoT ecosystem is diverse, consisting of devices with varying complexities in computation and communication. However, there is still a lack of consensus on standards and best practices among the companies to address the issue of interoperability. By the time, a standard would be agreed upon and adopted, current IoT deployments could become obsolete due to non-conformance of the policies outlined in the standard. Therefore, companies need to consider outfitting IoT deployments with mechanisms to ensure that these IoT deployments can be easily integrated with other existing and future systems as well as be able to implement any future standards, features, and services. The key to future-proofing and longevity of IoT deployments lies in hardware

flexibility. Reconfigurable processors can be used to impart flexibility in both processing and communication hardware in IoT deployments. Hardware reconfiguration enables IoT deployments to interoperate with disparate existing systems. Hardware reconfiguration also enables IoT deployments to be reprogrammed to fit any future standards or to implement new features and services.

In this paper, we propose the design of a flexible, high-performance, energy-efficient, secure, and dependable processor architecture for future IoT deployments. We propose a two-tiered heterogeneous and reconfigurable processor architecture that consists of a high-performance host processor, comprising of reconfigurable computation and communication units, which controls a number of low-power interface processors. The two-tiered heterogeneous architecture enables effective energy management and reconfigurability adds further flexibility and energy savings. We also equip our proposed IoT architecture with security co-processor extensions that support various security primitives, such as encryption and decryption, key generation, integrity verification, and device authentication. Finally, the proposed architecture incorporates dependability through various hardware- and software-based fault tolerance methods.

Our main contributions are as follows:

- Proposal of a novel two-tiered heterogeneous reconfigurable IoT processor architecture that imparts energy efficiency, high-performance, flexibility, security, and dependability to meet the diverse application requirements.
- Proposal of *security* co-processor extensions that leverage hardware-based security approaches to support various security primitives including encryption and decryption, key generation, integrity verification, and device authentication.
- Classification of chip multiprocessor benchmarks into IoT-specific application categories and using these benchmarks and a design space exploration methodology to determine low-power and high-performance microarchitecture configurations for the IoT processor.
- Implementation of selected security and dependability primitives of our proposed IoT architecture on a Xilinx Spartan-6 field-programmable gate array (FPGA) and comparison with an optimized implementation on an ARM processor in terms of performance and energy efficiency.

II. Related Work

There are many articles released by processor and system-on-chip (SoC) design companies that outline techniques of increasing processing capabilities in IoT devices. These articles focus on selecting processors suitable for the type and size of workload for IoT devices and on designing low-power optimized processor architectures for IoT.

ARM proposed a processor architecture consisting of multiple homogeneous processors in a single IoT object each serving a different purpose [4]. ARM defined a system with three Cortex-M processors, one to handle network connectivity, one to manage interface with sensors and actuators, and one as a host processor controlling the other two. ARM stated that multiple processors are better for lowering power consumption in IoT objects since only the processor serving the current task would be in active mode while the rest would be in sleep mode. ARM also proposed a guide to selecting microcontrollers for IoT objects [5]. In this guide, ARM argued that high-end microcontrollers are suitable for IoT deployments for two reasons. Firstly, high-end microcontrollers complete processing tasks sooner and can enter sleep mode to conserve power and secondly, larger flash and RAM sizes available with high-end microcontrollers facilitate implementation of complex networking protocols without addition of any new processors in the system. These articles clearly demonstrate the need for having more power-optimized processors in IoT deployments.

Synopsys also proposed the use of multiple processors in IoT deployments [3]. Synopsys described the use of two-tiered processor architecture in IoT objects—ultra low power embedded processors used to interface with sensing elements to collect, filter and process data, and host processor used to manage low power embedded processors. The processor architecture proposed by Synopsys lowered power consumption by keeping power hungry host processor mostly in sleep mode, similar to the concept used by ARM. Synopsys also discussed optimization of processors using configurable hardware extensions for sensor applications [3]. Synopsys stated that adding custom hardware extensions for executing typical sensor functions reduces the processor cycle count required to execute sensor applications. The reduction in cycle count lowers energy consumption either by lowering the clock frequency and keeping the same execution time, or having the same power but shorter execution time.

Contemporary approaches in energy-efficient architectures focus either on computational or communication aspects. However, our proposed architecture simultaneously considers both computation and communication aspects for attaining higher performance in energy-constrained IoT devices. Our proposed architecture also takes into account the diversity in the IoT domain with regards to devices having varying complexity. The flexibility in our proposed architecture makes it suitable for IoT devices with different energy and cost constraints. Overall, our proposed architecture presents a promising solution for meeting performance, real-time, energy, throughput, latency, and resilience requirements of IoT applications in a distributed heterogeneous IoT environment.

III. Reconfigurable IoT Architecture

A two-tiered heterogeneous processor architecture is suitable for increasing the processing capabilities of IoT while also maintaining energy efficiency [4] [6]. A high-level schematic of such a two-tiered architecture is shown in Figure 1. This architecture consists of a central host processor with a communication unit and a high-performance optimized computation unit. The host processor is interfaced with a number of low-power optimized interface processors. The interface processors carry out minor tasks, such as collecting data from sensors and

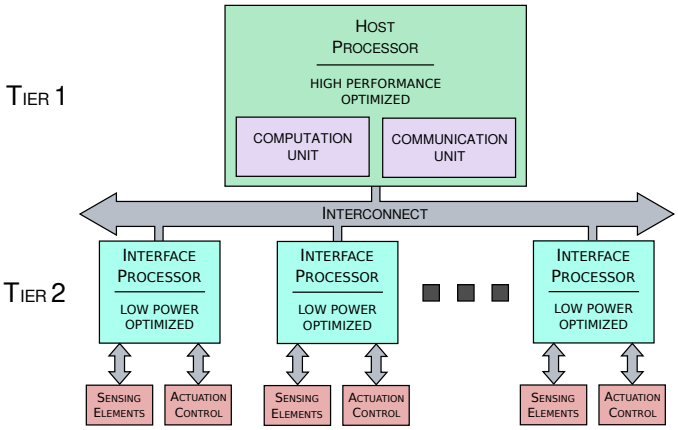

Fig. 1. Two-tiered heterogeneous processor architecture for IoT.

controlling actuation elements. The interface processors require minimal energy for operation so they do not significantly impact the battery life of IoT deployments. Hence, interface processors can always be operated in active mode. The host processor however expends a lot of energy during operation so it is only activated intermittently and for limited periods. The host processor is activated when compute-intensive functions, such as data-analysis, filtering, and complex security protocols need to be performed.

Figure 2 shows the details of our proposed two-tier heterogeneous processor architecture for IoT. Our proposed architecture is able to dynamically reconfigure both computation and communication parameters to attain high performance and energy efficiency. In the following subsections, we briefly describe the important components of our proposed IoT processor architecture shown in Figure 2 and discuss how these components contribute to improving overall performance and energy efficiency.

A. Host Processor

The host processor of our proposed architecture consists of a computation unit, a communication unit, and storage unit(s). Fault tolerance (FT) is provided for different tasks executed by the application and reconfigurable processors on a need basis depending on the criticality of function and IoT application. The host processor allows for reconfiguration of selected parameters (e.g., core count, operating frequency, modulation power, baseband filtering, etc.) of computation and communication units. The reconfiguration enables the host processor to add processing capabilities to IoT devices for mission-critical and/or emergency situations and remove the added capabilities to switch to an energy-efficient configuration in idle and/or regular operating situations.

1) Computation Unit: The computation unit within the host processor consists of an application processor alongside a reconfigurable processing unit that houses a reconfigurable processor and co-processor extensions.

Application Processor: The application processor in the computation unit of the host processor operates during idle/normal operating situations. When compute-intensive or application-specific tasks have to be performed, then the application processor is tasked with performing dynamic reconfiguration of the reconfigurable processing unit to create new processor and co-processor instances.

Reconfigurable Processor: Reconfigurable processor instances are engendered when heavy workloads need to be processed. The processor instances are generated by setting the values for a number of tunable processor parameters, such as core count, operational frequency, cache subsystem, instruction fetch/issue/retire widths, reorder buffer size, branch prediction, etc. The choice of values for these processor parameters is made based on the type and size of the workload. These processor instances are removed when they are not required during idle/normal operation to help improve energy efficiency.

Co-processor Extensions: The host processor in our proposed architecture consists of a number of co-processor extensions to aid in dedicated application-specific tasks. We provide a brief description and uses of these co-processor extensions in the subsections below:

Security: Since IoT devices are used for sensing and actuating applications, the IoT devices are the primary interface between the digital and the physical world. If an IoT node is compromised then an attacker acquires the ability to control the physical environment wherein the IoT device is deployed. For example, consider an industrial IoT deployment which is used to maintain the temperature of a warehouse at a certain limit. If an attacker gains direct access to an IoT node in the warehouse, then s/he can manipulate it to raise/lower the temperature inside the warehouse beyond the specified limit leading to damage of goods stored in the warehouse. Hence, it is crucial that IoT deployments have strong security features to protect against adversarial attacks. In order to implement strong security primitives, higher computational capabilities are required. To address the need for integrating strong security features in the IoT devices and higher computation ability required to implement these security features, we incorporate security co-processor extensions in our proposed heterogeneous processor architecture for IoT. The security co-processor extensions aid in implementing strong cryptographic primitives to secure data communicated to and from IoT devices.

Figure 3 shows a high level overview of the security primitives that are provisioned by the security co-processor extensions. These primitives enable confidentiality, integrity and authentication in our proposed architecture. We propose hardware acceleration for encryption/decryption operations (e.g., advanced encryption standard (AES)) and message authentication operations (e.g., hash-based message authentication code (HMAC)). Having hardware acceleration for these complex security primitives significantly improves performance and reduces energy cost [7]. Since encryption/decryption and message authentication are key-based primitives, a secret key is required for executing these security computations. In our proposed architecture, instead of storing secret keys in some on-chip storage element, we include a key generation module that is based on weak physically unclonable functions (PUFs) [8]. The PUF-based key generation module eliminates the need for having costly on-chip temper resistant memory. PUF-based

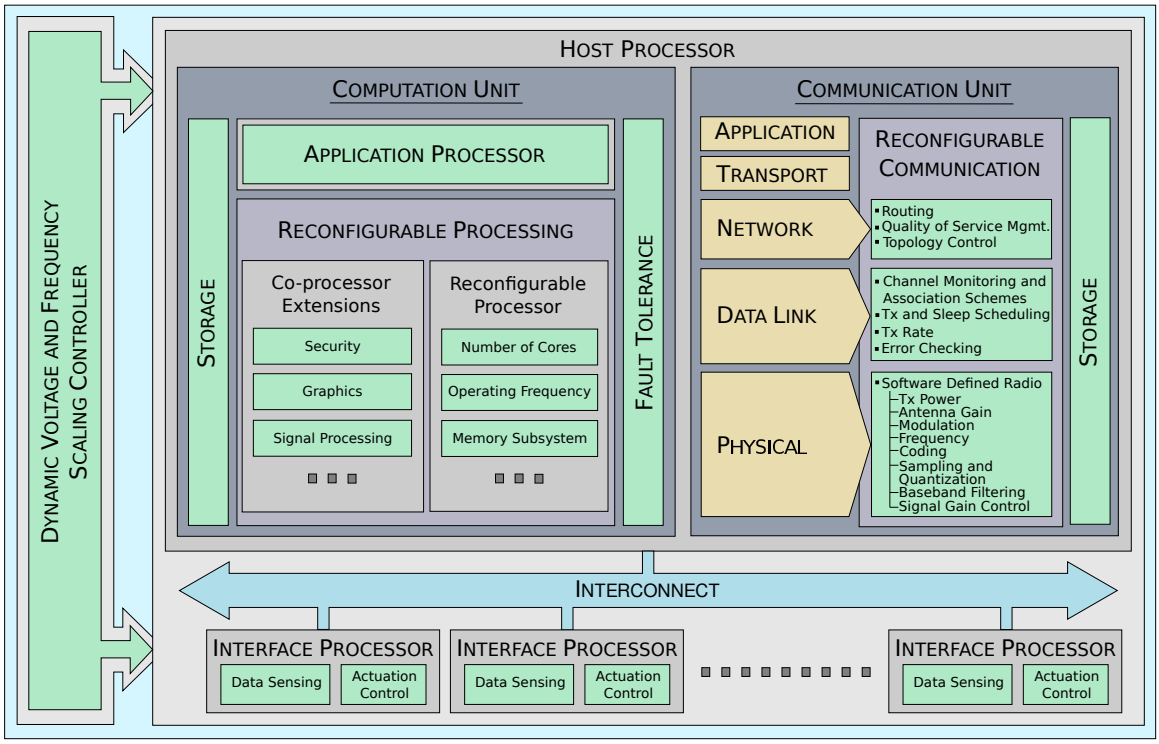

Fig. 2. Heterogeneous reconfigurable processor architecture for IoT including details of host and interface processors.

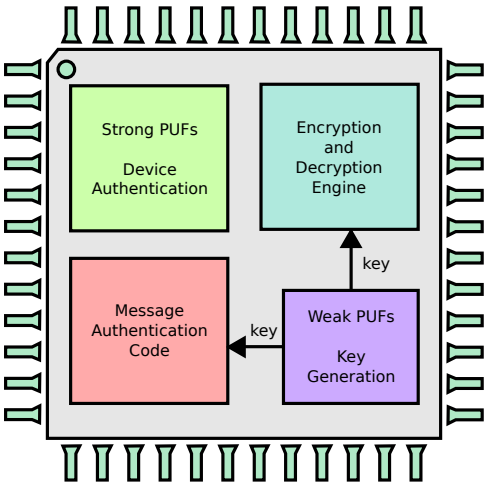

Fig. 3. Overview of security primitives in the proposed heterogeneous reconfigurable IoT architecture.

key generation provides a large key space within a smaller footprint as compared to implementing a tamper-resistant memory for secret key storage. For device authentication, we include a strong PUF-based authentication module. Strong PUF-based authentication schemes are discussed in [8].

Graphics: IoT deployments, such as surveillance and monitoring systems, have to collect and process a large amount of image data in order to carry out their assigned tasks. Image processing operations are compute intensive and highly data-parallel in nature. They require specialized high performance computing resources to operate on multiple number of similar threads in parallel. To provide such specialized support for image-based application domains, we incorporate graphics co-processor extensions in our proposed IoT processor architecture. The graphics co-processors provide support for simple image analysis operations like image segmentation, edge detection, motion detection, etc. Having specialized graphics hardware on-board lowers the execution time of graphics-related operations, thereby reducing the amount of time the host processor has to remain in an active state, which in turn helps to improve the energy efficiency of the IoT deployments.

Signal Processing: While performing sensing and actuation tasks, IoT devices have to convert signals between analog and digital domain. In the IoT domain, signal processing finds applications in various tasks, such as speech recognition, image compression, audio playback, etc. Our proposed IoT processor architecture includes a signal processing co-processor extension to provide dedicated support for signal processing applications, such as signal filtering, processing and transformation functions. The availability of specialized hardware for signal processing improves the performance of the host processor and contributes significantly to maximizing its energy efficiency.

2) Communication Unit: The communication unit is used to communicate data with other IoT devices or with computing nodes that are higher up in the network hierarchy. Our proposed architecture empowers reconfiguration in the communication unit to enable an IoT device to communicate with other IoT devices in the heterogeneous IoT environment wherein the devices use different communication architectures and networking protocols. The reconfigurability for communication unit includes modification of radio settings (e.g., transmission power, antenna gain, modulation, frequency, coding,

sampling and quantization, baseband filtering, and signal gain control), data link layer parameters (e.g., channel monitoring and association schemes, transmission and sleep scheduling, transmission rate, and error checking), and network layer parameters (e.g., routing, quality of service management, and topology control).

3) Storage: In our proposed IoT architecture, storage units are present in both the computation and communication units within the host processor. The storage unit within the computation unit is used for storing data for a variety of purposes, such as aggregation, analytics, mining, and archival. Interface processors gather data from different sensing elements and store that data in the storage unit. When complex data operations, such as filtering, sorting, etc., needs to be performed on the aggregated data, then the host processor reads data from the storage unit and operates on it. The storage unit also holds reconfiguration binaries that are used to perform dynamic reconfiguration of modules within the reconfigurable computation unit. The storage unit further stores locally relevant historical data which can be utilized by the host processor for analytics and making control decisions. The storage unit within the communication unit holds configuration parameters for software defined radios, network topology information, etc.

4) Fault Tolerance: Faults can result in IoT devices during normal operation due to environmental fluctuations (jitters, noise, radiations, etc.) or due to aging. The effects of faults on IoT devices can be mitigated by designing IoT devices to be fault tolerant. Fault tolerance is particularly important for IoT devices deployed for mission- and safety-critical applications. Fault tolerance can be provided through both hardware and software methods. The fault tolerance techniques employed by our proposed IoT architecture include: (i) fault tolerance by redundant multithreading (RMT), referred to as FT-RMT; (ii) FT-RMT enhanced with quick error detection (QED) [9]; (iii) dual modular redundancy (DMR); (iv) Berger code based totally self-checking combinational circuit (TSC); and (v) fault tolerance using self-reconfiguration in DMR (FT-SR-DMR). FT-SR-DRM performs dynamic self-reconfiguration to replace the faulty instances of the hardware module/component with new instances by exploiting partial reconfiguration feature of Xilinx Spartan-6 FPGA.

B. Interface Processors

The interface processors are optimized for low-power operation and are tasked with controlling interface components, such as sensors and actuators. Reading data from sensors and sending control actions to actuators has to be performed in short regular intervals, and thus require an always active processor. The low-power interface processors are well suited for these sensing and control applications because keeping these interface processors in active mode has minimal effect on the battery life of an IoT device.

C. Dynamic Voltage and Frequency Scaling Controller

Our proposed architecture also incorporates a dynamic voltage and frequency scaling (DVFS) controller that adjusts the operating voltage and frequency of various hardware components for meeting performance requirements while conserving energy. Voltage and frequency scaling is carried out in both the host and the interface processors to further improve the energy efficiency of our proposed architecture.

IV. Methodology and Experimental Setup

In this section, we describe the experimental setup for two independent set of experiments that we have performed for our proposed IoT architecture. In our first set of experiments, we use a design space exploration method for microarchitecture parameter tuning to determine microarchitecture parameters for high-performance optimized host processor and low-power optimized interface processor(s). In our second set of experiments, we implement and compare selected security and dependability primitives and compare the result in terms of performance and energy efficiency. The methodology and experimental setup for these experiments are described below.

A. Determining microarchitecture configurations using design space exploration

In order to optimize the microarchitecture parameters of the host and interface processors used in our proposed heterogeneous IoT architecture, we employ a design space exploration methodology that we have detailed in our previous work [10]. We utilize a four phase exploration algorithm consisting of the following phases: initial one-shot optimization and parameter significance phase, set partitioning phase, exhaustive search phase, and greedy search phase. We run our experiments on a cycle accurate multiprocessor simulator called ESESC (Enhanced Super ESCalar) [11] and use a set of PARSEC (Princeton Application Repository for Shared-Memory Computers) and SPLASH-2 (Stanford ParalleL Applications for SHared memory, version 2) benchmarks [12] [13] [14] to provide test workloads of varying types and sizes. We use a weighted objective function for ranking the different microarchitecture configurations tested by our search algorithms. The objective function is a weighted sum of the total power and total execution time design metric values that are obtained from simulation. The design space for the host and interface processors is shown in Table I.

B. Security and dependability approaches

For verification of performance and energy-efficiency of security primitives afforded by our proposed IoT processor architecture, we implement AES-128 for rendering confidentiality (encryption and decryption operations) and secure hash algorithm (SHA) based HMAC for message integrity verification. We have implemented the following dependability primitives as outlined in Section III-A4: FT-RMT, FT-RMT-QED, and FT-SR-DMR. We test two software-based implementations in this experiment. The first is baseline design (BD) that implements AES-128 and SHA-2 and has no code optimizations. The second is optimized baseline design (OptBD) that implements AES 128 and SHA-3 and incorporates code optimizations such as loop unrolling, cache-aware programming, alignment of data structures to cache line boundary, etc. Both BD and OptBD are implemented on a 32-bit quad-core Cortex-A9 ARM application processor processor running Ubuntu 14.04.4 LTS at 396 MHz clock speed. Our

TABLE I
DESIGN SPACE FOR MICROARCHITECTURE CONFIGURATION PARAMETERS FOR HOST AND INTERFACE PROCESSORS

Parameter Name	Set of Settings	
	Low-Power	High-Performance
Cores	1, 2, 4	2, 4, 8
Frequency (MHz)	75, 100, 125, 150	1700, 2200, 2800, 3200
L1-I Cache Size (kB)	8, 16, 32, 64	8, 16, 32, 64, 128
L1-D Cache Size (kB)	8, 16, 32, 64	8, 16, 32, 64, 128
L2 Cache Size (kB)	256, 512, 1024	256, 512, 1024
L3 Cache Size (kB)	2048, 4096	2048, 4096, 8192

TABLE II
CATEGORIZATION OF TEST BENCHMARKS ACCORDING TO IOT APPLICATIONS

IoT Application	Benchmarks
Data sensing and aggregation	Cholesky, Radix
Data analysis and Data mining	Blackscholes, Freqmine
Graphics	Facesim, Fluidanimate
Signal processing and Communication	FFT

proposed IoT processor architecture implements AES-128 and SHA-3 on a Xilinx Spartan-6 FPGA. We refer to the implementation of our IoT processor architecture on FPGA as ITAF. The ITAF incorporates dependability by implementing FT-SR-DMR (Section III-A4). We also compare FT and non-fault-tolerant (NFT) implementations in terms of performance and energy efficiency. Our previous work [7] provides further details on the implementation.

V. RESULTS

In this section, we present the results for the two sets of experiments outlined in Section IV.

A. Microarchitecture configurations obtained from design space exploration

IoT devices operate on a wide variety of workloads of different types and sizes. We broadly separate these workloads into four different categories relating to common IoT applications or processes. We classify the benchmarks from the PARSEC and SPLASH-2 benchmark suites into these categories based on the closest IoT application that the benchmarks resemble. The categorization of some of the key test benchmarks used in our experiments are shown in Table II.

1) Microarchitecture configurations for low-power optimized processors for IoT: Table III shows the microarchitecture configurations obtained for the Cholesky benchmarks from the SPLASH-2 benchmark suite. We use this as an example to discuss the microarchitecture configuration required in low-power optimized interface processors. We note that for the Cholesky benchmark, our design space exploration methodology selects the lowest operating frequency (75 MHz) and core count (single-core). This is because high operating frequency and high number of cores in the processor directly influences the power consumption of the processor. The size of the L1-D cache in the resulting configuration is also large because of the large workload offered by the Cholesky benchmark. This is representative of the growing IoT ecosystem in which large volumes of data are gathered from a large number of sensing elements. In Table III, we have also included the values for total power and execution times returned from the simulations. The power value ranges in the order of a few hundred milliwatts (mW) and the execution time ranges in the order of a few hundred milliseconds(ms). We observe that the resulting design heavily favors low-power consumption by sacrificing performance (high total execution time). The low-power usage of this microarchitecture makes it suitable for use as interface processor which can remain in active mode indefinitely without significantly hampering the energy budget of IoT devices.

TABLE III
MICROARCHITECTURE CONFIGURATIONS FOR LOW-POWER OPTIMIZED AND HIGH-PERFORMANCE OPTIMIZED PROCESSORS FOR IOT

Parameter Name	Microarchitecture Configurations	
	Low-Power	High-Performance
	Cholesky	Blackscholes
Cores	1	8
Frequency (MHz)	75	3200
L1-I Cache Size (kB)	8	64
L1-D Cache Size (kB)	32	128
L2 Cache Size (kB)	256	256
L3 Cache Size (kB)	2048	8192
Total Power (W)	0.0934	4.549
Execution Time (ms)	327.958	28.1239

2) Microarchitecture configuration for high-performance optimized processors for IoT: Table III shows the microarchitecture configurations obtained for Blackscholes benchmarks from the PARSEC benchmark suite. We use this as an example to discuss the microarchitecture configuration required in high-performance optimized host processors. We observe that for the Blackscholes benchmarks, which is classified under data analysis and data mining category in Table II, performance improvement is achieved through higher operating frequency (3200 MHz) and core count(8-cores). The size of the L1-D cache and L2 cache for this microarchitecture configuration is also high because Blackscholes is a highly data-parallel benchmark. This is typical of the type of data analysis tasks that need to be performed on data aggregated from each sensing element in an IoT device. Since the host processor is equipped with reconfigurable computation unit, the core count and the operating frequency can be dynamically altered. The resulting design, in this case, heavily favors performance (total execution time) over total power. The total power value is in the range of a few watts whereas the execution time is in the range of few tens of milliseconds. As the power requirement for the host processors is high, as shown by this example microarchitecture configuration, it must be mostly kept in the sleep mode and only be activated infrequently and for short periods of time. This is necessary to conserve the battery life of the IoT devices. However, since these processors have shorter execution times, the processors have to remain active for a shorter period of time to complete their designated tasks as compared to the processors not optimized for high performance.

B. Comparison of Security and Dependability Primitives

In this section, we present the results for performance (time in μs) and energy (μJ) for completing one AES encryption

TABLE IV
PERFORMANCE AND ENERGY RESULTS FOR BD, OPTBD, AND ITAF.

Operational Mode	Baseline Design (BD)			Optimized Baseline Design (OptBD)			FPGA Implementation (ITAF)		
	FT Mode	Time (μs)	Energy (μJ)	FT Mode	Time (μs)	Energy (μJ)	FT Mode	Time (μs)	Energy (μJ)
NFT	x	257	13.137	x	189	9.661	x	4.90	2.170
FT	FT-RMT	411	21.010	FT-RMT	207	10.581	FT-SR-DMR	6.53	6.647
	FT-RMT-QED	589	30.109	FT-RMT-QED	313	16.000			

computation plus one HMAC computation for BD, OptBD, and ITAF.

1) Timing Analysis: Table IV shows the timing performance of BD, OptBD, and ITAF. Comparison of BD and ITAF reveals that NFT ITAF is 52.45× faster than NFT BD. Furthermore, after embedding FT in BD by FT-RMT and in our FPGA implementation by FT-SR-DMR, ITAF is 62.94× superior than BD. Lastly, ITAF with FT-SR-DMR provides a speedup of 90.19× over BD in FT-RMT-QED mode.

Comparison of ITAF and OptBD shows that NFT ITAF is faster than NFT OptBD by 38.57×. Moreover, FT-SR-DMR in ITAF surpasses FT-RMT in OptBD by 31.69×. Furthermore, a speedup of 47.93× is achieved with FT-SR-DMR in ITAF over OptBD with FT-RMT-QED.

2) Energy Analysis: Table IV depicts the energy consumption results of our implementations of security and dependability primitives. The comparison between ITAF and BD reveals that NFT ITAF is 6.05× more energy efficient than NFT BD. ITAF with FT-SR-DMR is 3.16× more energy efficient than BD with FT-RMT and 4.52× more energy efficient than BD with FT-RMT-QED.

The comparison between ITAF and OBD shows that NFT ITAF results in 4.45× more energy savings than NFT OptBD. Additionally, ITAF with FT-SR-DMR gives 1.59× more energy savings than OptBD with FT-RMT, and 2.4× times more energy savings than OptBD with FT-RMT-QED, respectively.

VI. CONCLUSIONS

In this paper, we have proposed the design of a novel two-tiered heterogeneous processor architecture for IoT that imparts energy efficiency, high-performance, flexibility, security, and dependability to meet the diverse application requirements. Our proposed architecture consists of a high-performance optimized reconfigurable host processor that controls a number of low-power optimized interface processors. We utilize a design space exploration methodology for processor parameter tuning, using a cycle-accurate simulator (ESESC) and a standard set of PARSEC and SPLASH-2 chip multiprocessor benchmarks, to determine example microarchitecture configurations for the host and interface processors. From the resulting microarchitecture configurations, we observe that the high-performance optimized host processor requires a higher core count and operating frequency as compared to the low-power optimized interface processor. The size of the different levels of caches in the microarchitecture configuration depends on the size of the workload. Results indicate that the resulting microarchitecture configurations for both the host processor and the interface processor possess large cache size.

In this paper, we have also implemented selected security and dependability primitives of our proposed IoT architecture on a Xilinx Spartan-6 FPGA and have compared the results with baseline and optimized implementations on an ARM processor in terms of performance and energy efficiency. Experimental results show that the FPGA-prototype implementations of security and dependability primitives of our proposed IoT processor architecture outperform ARM-based implementations by 47.93× while consuming 2.4× lesser energy. These results support our concept for including hardware-based security co-processor extensions in the host processor of our proposed architecture. As our future work, we plan to prototype additional features of our proposed security coprocessor extensions, such as device authentication and key generation using PUFs.

REFERENCES

[1] J. Chase, "The evolution of the internet of things - from connected things to living in the data, preparing for challenges and IoT readiness," Texas Instruments, Tech. Rep., Sep 2013.

[2] "What the internet of things (IoT) needs to become a reality," Freescale, Tech. Rep., May 2014.

[3] J. Geuzebroek and A. Vaassen, "Building an efficient, tightly coupled embedded system using an extensible processor," Synopsys, Tech. Rep., Jun 2014.

[4] "Intelligent flexible IoT nodes," ARM, Tech. Rep., Oct 2015.

[5] K. Char, "Internet of things system design with integrated wireless MCUs," Silicon Labs, ARM, Tech. Rep., Oct 2015.

[6] S. Bath. (2016, Aug) Developing solutions for the internet of things. [Online]. Available: https://www.intrinsyc.com/increasing-solution-differentiation-edge-based-heterogeneous-computing/

[7] B. Poudel and A. Munir, "Design and evaluation of a novel ecu architecture for secure and dependable automotive cps," in *2017 14th IEEE Annual Consumer Communications Networking Conference (CCNC)*, Jan 2017, pp. 841–847.

[8] C. Herder, M.-D. Yu, F. Koushanfar, and S. Devadas, "Physical unclonable functions and applications: A tutorial," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1126–1141, Aug 2014.

[9] T. Hong, Y. Li, S.-B. Park, D. Mui, D. Lin, Z. A. Kaleq, N. Hakim, H. Naeimi, D. S. Gardner, and S. Mitra, "QED: Quick Error Detection Tests for Effective Post-Silicon Validation," in *Proc. of IEEE Internation Test Conference (ITC)*, Austin, Texas, November 2010.

[10] P. Kansakar and A. Munir, "A design space exploration methodology for parameter optimization in multicore processors," *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 1, pp. 2–15, Jan 2018.

[11] E. K. Ardestani and J. Renau, "ESESC: A fast multicore simulator using time-based sampling," in *Proceedings of IEEE 19th International Symposium on High Performance Computer Architecture (HPCA)*, Washington, DC, USA, Feb 2013.

[12] C. Bienia, "Benchmarking modern multiprocessors," Ph.D. dissertation, Department of Computer Science, Jan 2011.

[13] Y. Bao, C. Bienia, and K. Li, *The PARSEC Benchmark Suite Tutorial - PARSEC 3.0*, San Jose, CA, USA, Jun 2011.

[14] S. C. Woo, M. Ohara, E. Torrie, J. P. Singh, and A. Gupta, "The SPLASH-2 programs: Characterization and methodological considerations," in *Proceedings of 22nd Annual International Symposium on Computer Architecture (ISCA)*, Santa Margherita Ligure, Italy, Jun 1995.