Multi-Core Data Analytics SoC with a Flexible 1.76 Gbit/s AES-XTS Cryptographic Accelerator in 65 nm CMOS

Frank K. Gürkaynak
Integrated Systems Laboratory,
ETH Zürich
CH-8092, Switzerland
kgf@ee.ethz.ch

Robert Schilling
Institute for Applied Information Processing and Communications
TU-Graz
8010, Austria
robert.schilling@iaik.tugraz.at

Michael Muehlberghuber
Integrated Systems Laboratory
ETH Zürich
CH-8092, Switzerland
mbgh@iis.ee.ethz.ch

Francesco Conti
Energy Efficient Embedded Systems Laboratory
University of Bologna
40135, Italy
f.conti@unibo.it

Stefan Mangard
Institute for Applied Information Processing and Communications
TU-Graz
8010, Austria
stefan.mangard@iaik.tugraz.at

Luca Benini
Integrated Systems Laboratory
ETH Zürich
CH-8092, Switzerland
lbenini@iis.ee.ethz.ch

ABSTRACT

Embedded systems for Internet-of-Things applications present new challenges to system design. From a hardware design perspective, energy efficiency is paramount, as most of devices have a limited power supply due to size considerations. Transmitting data away from the node remains a very power hungry operation, and the only viable solution to this problem is to reduce the amount of data by performing pre-processing which again requires additional computational power. Hence modern embedded devices need to strike a fine balance between the power needed for acquisition/processing and communication. In many scenarios, small IoT devices will be deployed widely making them vulnerable to malicious attacks. Thus, for practical applications, these devices also need to fit the necessary resources to provide adequate security services.

We present a cryptographic hardware accelerator capable of supporting multiple encryption and decryption modes for different cryptographic algorithms (AES, Keccak) in an energy efficient multi-core cluster optimized for embedded digital signal processing applications implemented in 65 nm CMOS technology. We show that it is possible to have the necessary computation power to perform cryptographic services in addition to state of the art processing in a power budget that is compatible with IoT devices in a mature 65 nm CMOS technology. When running at 0.8 V the SoC with the cryptographic accelerator can be clocked at 84 MHz running AES-XTS at more than 250 Mbits/s consuming a total of 27 mW, which is a 100 × gain in energy and 496 × gain in operation speed over an optimized software implementation running on a single 32 bit OpenRISC core.

CCS Concepts

•Security and privacy → Embedded systems security; Hardware security implementation; •Hardware → Very large scale integration design; Application specific processors;

Keywords

VLSI design; Hardware Cryptography; Embedded Systems; AES

1. INTRODUCTION

State of the art embedded systems can be manufactured with small form factors allowing them to be included in everyday items. On their own, such small systems may be used to collect data from a modest set of sensors, but when aggregated together in a network, such Internet-of-Things applications have the potential to unlock many new and exciting services. The key to such applications is to have individual nodes that can be active for long periods of time without manual intervention, which requires an adequate power supply for the expected lifetime. Since form factor and weight considerations severely limit the battery capacity, such systems must simply learn to do more with the available energy.

A system used for IoT applications needs energy to sample signals, optionally process these signals and transmit these over a communication channel. By far the largest bottleneck in this system lies in the communication overhead [14], where the only realistic approach is to reduce the amount of data that is transmitted. Significant amounts of data reduction can only be achieved, by nodes that can make *sense* of their environment [11]. For example rather than transmitting a still image, the node might already detect relevant features in the image and only transmit recognized features. This, in

turn, requires significant computing capability which taxes the power budget.

Application specific circuits deliver the most energy efficient computing solutions in this domain. However, such circuits are customized for one application and offer very little flexibility. This is a huge problem for the IoT field, where new applications are constantly invented and refined. Such applications require the flexibility that can only be satisfied by programmable devices. Recent developments have yielded microcontrollers with very high energy efficiency [10], but at the moment, these systems lack the performance to realize complex signal processing required to reduce the amount of data transmitted by an IoT node significantly [7]. Systems built around such microcontrollers, still rely on transmitting the data they have captured directly to a centralized device, usually a smartphone, using a significant portion of their battery for this purpose and limiting their application.

Security is one of the main problems faced by IoT systems. By design, IoT nodes are destined to be deployed widely and could easily be captured and or monitored by adversaries. In addition, as these devices are resource constrained, equipping such nodes with necessary tools to provide reasonable security is more challenging. An IoT device will not only need bulk encryption to ensure confidentiality of the data that it puts on the communication channel but will require a number of different cryptographic services, to establish session keys, provide authentication, etc. Considering that most IoT end devices will not be deployed in physically secure environments, IoT devices are especially vulnerable to side-channel attacks. To counter such attacks, cryptographic algorithms augmented with countermeasures against side-channel attacks are needed. However, these countermeasures come with a significant workload of 100-1000s of cycles per encrypted byte [3]. Finally, smart IoT nodes that make *sense* of the environment contain data that has been *distilled*, which potentially carries more value to both the user and adversaries and therefore will need better protection than data collected by IoT nodes that simply relay the data.

In this paper, we share the measurement results from a multi-core SoC for embedded IoT applications that has been augmented with two accelerators, one to increase the computational capabilities for feature extraction using convolutional neural networks (CNNs), and the second one to provide the systems with a set of cryptographic primitives to provide different cryptographic services. We demonstrate that it is possible to perform cryptographic operations at a rate that would allow them to be used for advanced functionality such as memory/storage encryption even in the most resource constrained environments.

2. PULP PLATFORM

The energy efficiency of standard CMOS circuits has been well researched. It is a well-known fact that CMOS circuits are more energy efficient in the so-called Near-Threshold region [4]. While circuits operating in this region are more efficient and do more per amount of energy consumed, they operate at a lower speed, decreasing the throughput of the system. A promising approach has been to use parallelization to offset this disadvantage and design near-threshold multi-core systems with very high energy efficiency when coupled with traditional power and energy saving methods

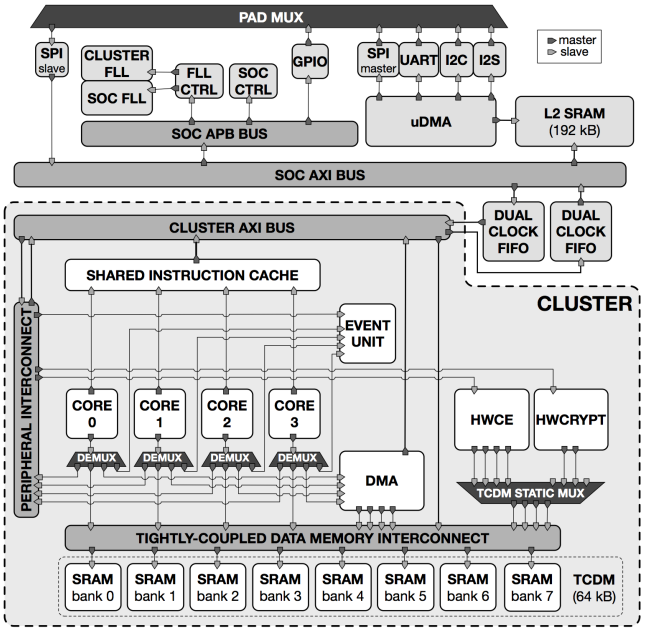

Figure 1: Block diagram of the Fulmine SoC, A multi-core cluster with four OpenRISC cores, and two hardware accelerators for cryptography (HWCRYPT) and convolutional neural networks (HWCE).

such as dynamic voltage and frequency scaling and power and clock gating. We have been developing an open[1] Parallel Ultra-Low-Power (PULP) Platform [13] on these principles.

In this paper, we present a 3rd generation PULP platform SoC called *Fulmine* that consists of two parts with independent voltage and frequency domains communicating through an AXI4 interconnect and separated by dual-clock FIFOs and level shifters. The *cluster* (shaded part in Figure 1) contains four 32 bit OpenRISC cores and two accelerators, while the *SoC* part contains a 192 KB L2 memory and peripheral interfaces including a micro-DMA subsystem that can directly copy data between external interfaces (QSPI, I2C, I2S, UART) and the L2 memory, while the cluster is in a sleep state. Aggressive clock gating ensures that inactive parts of the system are put into sleep mode and a programmable event unit is used to wake up cluster modules when necessary. Two frequency-locked loops (FLLs) generate clocks for the cluster and the rest of the SoC, while at the moment the two domains rely on external voltage regulators for their power supply.

The heart of the system is the cluster containing the computation modules. A key part of the system is the Tightly Coupled Data Memory (TCDM) that connects eight banks of SRAM macros over a logarithmic interconnect providing a total of 64 KB shared memory accessible to all computing blocks [12]. This organization allows all blocks parallel single cycle memory access unless there are contentions. In case of contentions, a starvation-free round-robin arbitration policy is used to stall all but one of the memory access requests. The interconnect structure and the banking ensure that conflicts are reduced to a minimum, and in practice do not exceed 10% even for the most load/store intensive ap-

[1]Available online at http://pulp-platform.org

plications. Data is copied to and from the TCDM using a dedicated multi-channel DMA.

The cluster contains four 32 bit OpenRISC cores with custom SIMD extensions to allow vector processing for basic arithmetic operations [6]. Support for these extensions has been added to the GCC compiler allowing applications written in C/C++ to be directly ported to Fulmine. The cores fetch their instructions from a shared Instruction Cache which improves performance when executing parallel operations on data intensive kernels.

What is truly novel in the design of the cluster of Fulmine is the inclusion of two accelerators: a cryptographic accelerator (HWCRYPT) described in more detail in the following section, and a dataflow convolution-accumulation engine (HWCE) optimized for the execution of Convolutional Neural Networks (CNNs) over 16 bit fixed-point sensor data with 4, 8, or 16 bit weights. The HWCE can be configured to calculate convolutions using a window size of 5×5, or smaller and using a sliding window can compute one output feature with 16 bit, two features with 8 bit or four features with 4 bit weights with an average throughput of 0.87, 1.64, and 2.22 pixels/cycle respectively.

Fulmine is the result of ongoing research in our group and non-volatile memory IP for this technology was not available to us during the design. In a commercial setting, non-volatile memories would have been used for the L2 memory. A small ROM allows Fulmine to boot from an external SPI ROM during reset, and a standard JTAG interface can also be used to directly program the SoC.

3. CRYPTOGRAPHIC ACCELERATOR

The cryptographic accelerator HWCRYPT shown in Figure 2 is a dedicated functional unit designed to interface the shared memory architecture used in Fulmine and supports a variety of cryptographic primitive functions. The main datapath consists of two parallel units, one implementing the popular AES block cipher, and the other one implementing a sponge based Keccak-f400 (the algorithm that was selected as SHA-3) albeit with a smaller 400 bit internal state.

The AES engine in HWCRYPT includes two parallel instances each containing two rounds of the AES round supporting both encryption and decryption. Round keys are generated on the fly; an internal register is used to keep the last round key during encryption, which is then used as the starting point for generating roundkeys for decryption. HWCRYPT allows the user to perform a complete AES en/decryption over 10 rounds, but also supports modes where individual AES rounds can be executed. As several new cryptographic algorithms are derived from AES rounds, this feature could help accelerate new algorithms that make use of AES rounds. The real strength of the unit lies in the fact that several more advanced AES based operations can be performed directly. For example, HWCRYPT can realize the XTS (XEX-based tweaked-codebook mode with ciphertext stealing), a recommended mode of operation by the National Institute of Standards and Technology (NIST) [5] which is commonly used for disk encryption achieving 0.38 cpb (cycles per byte). This is almost twice as much when compared to the AES-NI extensions found in modern Intel processors which achieve about 0.65 cpb [2]. When running at the energy optimum operating point of 84 MHz at 0.8 V this results in a throughput of 1.76 Gbit/s when running AES-XTS.

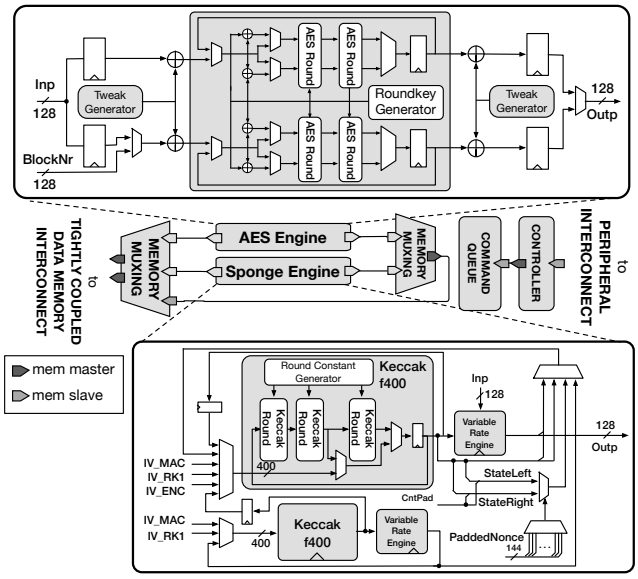

Figure 2: Top-level block diagram of the HWCRYPT cryptographic accelerator with two parallel cryptographic units.

The Sponge unit has two instances of the Keccak-f400 permutation each containing three rounds. The number of rounds were chosen to match the critical path of the AES unit. The sponge construction in HWCRYPT allows several trade-offs between security and performance. For example, the number of rounds used for one encryption operation as well as the rate of the system can be programmed. The rate in a sponge construction determines how many bits of the internal 400 bit state will be processed per round. This number can be chosen as a power of two between 1 (most secure) and 128 (fastest). The number of rounds can be chosen between 3 and 18 with increments of three as well as the 20 rounds as specified in the Keccak standard. The two Keccak instances can be combined to perform an authenticated encryption by using one unit for encryption and the other one for authentication. In this mode, HWCRYPT is able to achieve 0.51 cpb when operating on 8 kB blocks. The sponge unit also provides encryption without authentication and direct access to the permutations to allow the software to accelerate any KECCAK-f[400]-based algorithm.

The accelerator is controlled by a set of dedicated registers accessible by the processor cores that specify the location and the size of the source and destination memory regions as well as the function to be executed. The accelerator is then able to fetch the input data directly from the memory, perform the operation and write the result back to the memory where it will be available for further processing by the cluster. It can be configured to generate an event or interrupt to wake up components waiting for the result of the computation. The configuration interface also has a four stage command queue, which allows subsequent operations to be queued while the accelerator is still running an outstanding operation, eliminating configuration latency between subsequent operations.

As can be seen in Figure 1, HWCRYPT is connected to the TCDM using two ports, allowing 64 bit read or write access per cycle. The individual engines within the HWCRYPT

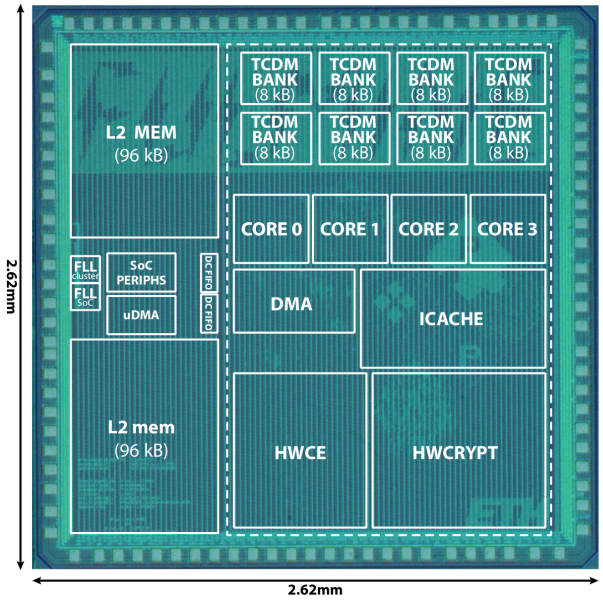

Figure 3: Annotated chip micrograph showing the Fulmine SoC. The main operational blocks have been highlighted.

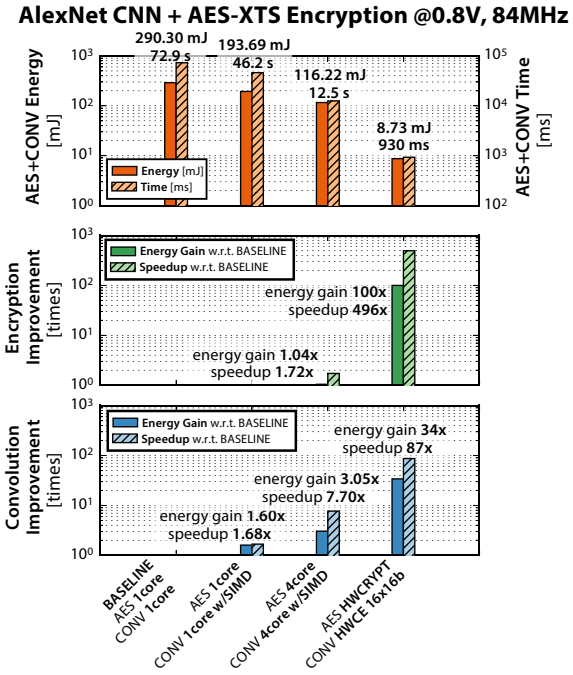

Figure 4: Measurement results showing improvement in energy consumption and execution speed of an application that uses AES-XTS encryption and Convolutional Neural Networks for different configurations of the SoC.

have been optimized to make the most of the available I/O bandwidth of the module. For example, a typical AES encryption requires 10 cycles and consumes and produces 128 bit. By implementing two rounds in hardware, HWCRPYT is able to compute this operation in five cycles and utilizes 80% of the available bandwidth.

In most scenarios, attackers would have relatively easy access to IoT devices and perform various side-channel attacks. Therefore it is important to have systems that can offer resistance against known side-channel attacks. Although HWCRYPT has been designed with side-channel resistance in mind and provides several leakage resilient cryptographic operation modes, the details of the countermeasures will not be presented in this paper, as the efficiency of these countermeasures are still being experimentally verified at the time of writing.

4. IMPLEMENTATION

We have implemented Fulmine using UMC 65nm LL 1P8M CMOS technology using a mixed VT synthesis technique to achieve a good balance between performance and energy consumption. Figure 3 shows a micrograph of the manufactured chip, where the approximate location and size of the main building blocks are highlighted. The chip occupies 2.62 mm × 2.62 mm including the I/O pads, and the net core area is 5.75 mm^2, of which 0.56 mm^2 is occupied by the HWCRYPT. The micrograph also highlights the cluster area (dashed rectangle) that occupies a separate power domain.

The manufactured chips were tested using our Advantest SoC V93000 ASIC tester, and Fulmine was shown to be operational from 1.2 V down to 0.8 V, at which point the larger SRAM macros we have used stopped functioning reliably. The highest energy efficiency, as expected, was achieved at 0.8 V where the entire chip was operational at 84 MHz. Tests were then carried out on dedicated development boards.

5. RESULTS

The HWCRYPT is able to accelerate basic cryptographic functions significantly. A straightforward Electronic Codebook (ECB) mode AES on 1 kB data blocks takes in the best case 344 cycles including all the setup and configuration including loading a fresh key to the HWCRYPT accelerator. This results in 0.34 cpb a speed-up of about 500 × when compared to an optimized AES-XTS implementation running on one 32 bit OpenRISC core. Even if a parallel AES-XTS implementation[2] running on four OpenRISC cores is used, the speed up is still more than 250 ×. These results can be seen in Figure 4 in the middle strip that compares the speed-up and energy gain for running AES-XTS in software on a single OpenRISC core (AES 1 core), in software running on four OpenRISC cores in parallel (AES 4 core) and running in the accelerator (AES HWCRYPT).

The more important metric for IoT applications is however energy consumption. While the entire system when running HWCRYPT at full speed consumes more power (27 mW at 0.8 V room temperature), overall the energy consumption also significantly decreases when shorter execution times are factored in. For the AES-XTS example shown in Figure 4, executing on HWCRYPT achieves a net energy gain of about 100 × over software implementations. Note that, Fulmine has been designed with fine-grained clock gat-

[2]The parallel implementation achieves 2 × speedup when four cores work in parallel on the same datablock. This implementation is necessary for operation modes that depend on the result of previous data blocks. For non-feedback modes like the AES-ECB mode, it is possible to have each core work on different data blocks resulting in 4 × speedup.

ing to turn off all unused hardware to reduce power consumption. Idle cores and accelerators, therefore, do not contribute to dynamic power consumption at all.

We have compared the performance of HWCRYPT with some of the recent cryptographic accelerators reported in literature in Table 1. Although the technology we have used is more mature than the state of the art, we are able to obtain higher performance and when accounting for the technology scaling better efficiency while supporting many additional cryptographic modes. Please note that the power number given is for the entire Fulmine SoC and not only the HWCRYPT unit and the efficiency numbers also contain the contribution of the rest of the system and include all the necessary setup operations as well as the memory transfers from TCDM. Even with all this overhead, the entire system supporting multiple modes consumes 15 pJ per encrypted bit which is only 5 × larger than an optimized dedicated-datapath supporting only AES encryption [1] showing that with Fulmine we were able to close the gap between programmable systems and dedicated hardware implementations. As noted earlier, the operation voltage for Fulmine was dictated by the minimum operation voltage of the SRAM macros that were available to us at design time. Our calculations show that the maximum energy efficiency to be between 0.55 V and 0.60 V for this technology, and we expect higher energy efficiency when SRAM macros that can operate at this voltage are used.

5.1 Secure Data Analytics Example

To present a more realistic application for Fulmine, we devised a scenario where the SoC would run the second convolutional layer (CONV2) of the AlexNet CNN [8], computing 256 feature maps from a set of forty eight 27×27 input channels with 5×5 filters. Partial results are stored on an external SD memory card using AES-128-XTS. In this scenario, HWCRYPT and HWCE work directly on the shared TCDM, allowing free data exchange with one another and with the cores, without copy overhead. Figure 4 compares secured AlexNet using scalar, SIMD-optimized, and parallel SIMD-optimized software against the two hardware accelerators. Using both accelerators, the test is executed in 485 ms spending 4.08 mJ of energy, speeding up the execution by 25.7 × and reducing the energy consumption by 28.5 × against parallel SIMD-optimized SW running on four cores. This example involves an equivalent workload of 4.05 billion OpenRISC instructions; using the hardware accelerators, the SoC consumes 4.08mJ, achieving an energy efficiency of 1.01 pJ per equivalent RISC instruction.

The relative contribution to the total power of individual modules while running the secured AlexNet example described above is given in Figure 5. At the minimum energy point (84 MHz at 0.8 V, room temperature) the power consumption of the entire SoC was measured to be 27 mW.

6. CONCLUSIONS

Near-sensor data analytics is a promising direction for IoT end-nodes, as it minimizes energy spent on communication and reduces network load. However, performing analytics (e.g. feature extraction or classification) directly on the end-nodes poses security concerns, as valuable data, *distilled* with application knowledge, is stored or sent over a network at various stages of the analytics pipeline. Protecting sensitive data at the boundary of the on-chip analytics engine is a way to address data security and privacy issues.

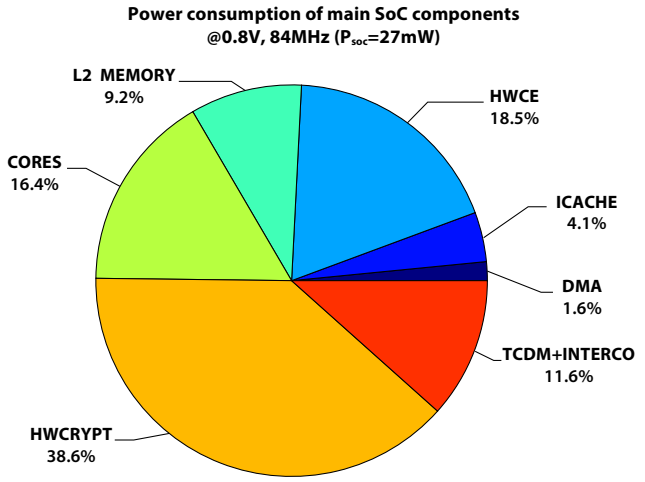

Figure 5: Power consumption of Fulmine SoC utilizing both cryptographic (HWCRYPT) and convolutional (HWCE) accelerators as described in the paper.

In this paper, we have presented a multi-core SoC suitable for IoT end-nodes, that has been augmented with specialized blocks for compute-intensive data processing and encryption functions, but still retains full software programmability for lower computational-intensity functions and provides configuration flexibility in interleaving cryptographic services at various levels of the analytics pipeline. We show that the proposed SoC shows similar efficiency as the state-of-the-art in AES encryption [9, 15] at 4 × the performance, while providing more cryptographic functionality. The multi-core architecture allows data to be transferred between different accelerators and cores without additional overhead using the shared TCDM approach. In an example application, Fulmine is able to make AES-128-XTS encryption, CNN-based analytics, and software-defined functions work together seamlessly, reducing the energy consumption of the complete example application by 28.5 × while remaining within a small power budget (27mW @ 0.8V).

Acknowledgments

This work was partially supported by the IcySoC RTD project evaluated by the Swiss NSF and funded by Nano-Tera.ch with Swiss Confederation financing and has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 681402).

7. ADDITIONAL AUTHORS

Pasquale Davide Schiavone - ETH Zürich
(email: `pschiavo@iis.ee.ethz.ch`),
Davide Rossi - University of Bologna
(email: `davide.rossi@unibo.it`),
Antonio Pullini - ETH Zürich
(email: `pullinia@iis.ee.ethz.ch`),
Michael Gautschi - ETH Zürich
(email: `gautschi@iis.ee.ethz.ch`),
Igor Loi - University of Bologna
(email: `igor.loi@unibo.it`),

Table 1: Comparing the performance of the encryption unit to state of the art.

	Our Work	Zhang et. al. [15]	Mathew et al. [9]	
Technology	UMC 65nm LL 1P8M	TSMC 40 nm	Intel 22 nm	
Operating Point	0.8 V, 84 MHz	0.9 V, 1.3 GHz	**0.9 V**, 1.13 GHz	**0.43 V**, 324 MHz
Area	0.56 mm^2 (HWCRYPT)	0.42 mm^2 (AES)	0.19 mm^2 (AES)	
Power	27 mW (SoC)	4.39 mW (AES)	13 mW (AES)	0.43 mW (AES)
Performance	1.76 Gbit/s	0.446 Gbit/s	0.432 Gbit/s	0.124 Gbit/s
Energy Efficiency	65.2 Gbit/s/W	113 Gbit/s/W	33.2 Gbit/s/W	289 Gbit/s/W
Supported Modes	AES-ECB, AES-XTS, Keccak-f400 (encryption, authentication)	AES-ECB	AES-ECB	

Germain Haugou - ETH Zürich
(email: haugoug@iis.ee.ethz.ch)

8. REFERENCES

[1] S. Banik, A. Bogdanov, and F. Regazzoni. *Exploring Energy Efficiency of Lightweight Block Ciphers*, pages 178–194. Springer International Publishing, Cham, 2016.

[2] A. Bogdanov, M. M. Lauridsen, and E. Tischhauser. *Comb to Pipeline: Fast Software Encryption Revisited*, pages 150–171. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015.

[3] D. Dinu, Y. L. Corre, D. Khovratovich, L. Perrin, J. Grossschadl, and A. Biryukov. Triathlon of lightweight block ciphers for the internet of things. Cryptology ePrint Archive, Report 2015/209, 2015. http://eprint.iacr.org/2015/209.

[4] 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*, 98(2):253–266, Feb 2010.

[5] M. Dworkin. Recommendation for Block Cipher Modes of Operation: The XTS-AES Mode for Confidentiality on Storage Devices. *NIST Special Publication*, 800, 2010.

[6] M. Gautschi, P. D. Schiavone, A. Traber, I. Loi, A. Pullini, D. Rossi, E. Flamand, F. K. Gurkaynak, and L. Benini. A near-threshold RISC-V core with DSP extensions for scalable IoT Endpoint Devices. *arXiv:1608.08376 [cs]*, Aug. 2016.

[7] M. Konijnenburg, S. Stanzione, L. Yan, D. W. Jee, J. Pettine, R. van Wegberg, H. Kim, C. van Liempd, R. Fish, J. Schluessler, H. de Groot, C. van Hoof, R. F. Yazicioglu, and N. van Helleputte. 28.4 a battery-powered efficient multi-sensor acquisition system with simultaneous ecg, bio-z, gsr, and ppg. In *2016 IEEE International Solid-State Circuits Conference (ISSCC)*, pages 480–481, Jan 2016.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.

[9] S. Mathew, S. Satpathy, V. Suresh, M. Anders, H. Kaul, A. Agarwal, S. Hsu, G. Chen, and R. Krishnamurthy. 340 mv-1.1 v, 289 gbps/w, 2090-gate nanoaes hardware accelerator with area-optimized encrypt/decrypt gf(2 4) 2 polynomials in 22 nm tri-gate cmos. *IEEE Journal of Solid-State Circuits*, 50(4):1048–1058, April 2015.

[10] A. Micro. Apollo datasheet ultra-low power mcu family. Technical Report, 2015.

[11] E. F. Nakamura, A. A. F. Loureiro, and A. C. Frery. Information fusion for wireless sensor networks: Methods, models, and classifications. *ACM Comput. Surv.*, 39(3), Sept. 2007.

[12] A. Rahimi, I. Loi, M. R. Kakoee, and L. Benini. A Fully-Synthesizable Single-Cycle Interconnection Network for Shared-L1 Processor Clusters. In *2011 Design, Automation & Test in Europe*, pages 1–6. IEEE, Mar. 2011.

[13] D. Rossi, F. Conti, A. Marongiu, A. Pullini, I. Loi, M. Gautschi, G. Tagliavini, A. Capotondi, P. Flatresse, and L. Benini. PULP: A parallel ultra low power platform for next generation IoT applications. In *Hot Chips 27 Symposium (HCS), 2015 IEEE*, pages 1–39. IEEE, 2015.

[14] V. Shnayder, M. Hempstead, B.-r. Chen, G. W. Allen, and M. Welsh. Simulating the power consumption of large-scale sensor network applications. In *Proceedings of the 2Nd International Conference on Embedded Networked Sensor Systems*, SenSys '04, pages 188–200, New York, NY, USA, 2004. ACM.

[15] Y. Zhang, K. Yang, M. Saligane, D. Blaauw, and D. Sylvester. A compact 446 gbps/w aes accelerator for mobile soc and iot in 40nm. In *2016 IEEE Symposium on VLSI Circuits (VLSI-Circuits)*, pages 1–2, June 2016.